

Toward data-driven materials design: From atoms to pilot plants

Présentée le 19 mai 2023

Faculté des sciences de base
Laboratoire de simulation moléculaire
Programme doctoral en chimie et génie chimique

pour l'obtention du grade de Docteur ès Sciences

par

Kevin Maik JABLONKA

Acceptée sur proposition du jury

Prof. A. Züttel, président du jury
Prof. B. Smit, directeur de thèse
Prof. J. Schrier, rapporteur
Prof. L. Gagliardi, rapporteuse
Prof. A. R. Natarajan, rapporteur

ABSTRACT

Discovering new materials is essential but challenging, time-consuming, and expensive. In many cases, simulations can be useful for estimating material properties. For many of the most interesting properties, however, simulations are infeasible because of prohibitive costs or because it is unknown how to set up a suitable simulation. A promising alternative to reduce the cost of predicting material properties—or to make estimates possible in the first place—is to learn mappings from materials or processes to properties from data. While the feasibility of this approach finds support in the observation that chemists learn from experience (intuition), such a data-intensive approach has unique challenges. First, it relies on suitable data to enable the research. Second, appropriate tooling is needed to enable a scientific research approach in which research findings can be easily compared and reused. A third challenge is that crystal structures often must be first converted into suitable inputs for machine learning models (so-called featurization).

In the first part of this thesis, we present tools that address all these challenges. Using open-source electronic lab notebooks, we capture data in a machine-actionable form. We then show how to use tooling from our “ecosystem for reticular chemistry”, which provides datasets, data splitters, featurizers, and benchmark utilities, to build, compare and publish machine learning models. A subsequent chapter then highlights how such machine-learning models can guide which experiments or simulations to perform next, particularly in the multiobjective setting, which is relevant for most material design problems.

The second part of this thesis uses tools from this toolbox for data-driven research to address problems from the atom to the pilot plant scale using a data-driven approach. On the atom scale, we show that chemically sensible features can be used to predict oxidation states of metal cations—a property at the heart of chemistry but not a quantum-mechanical observable. On the pilot plant scale, we address how a carbon-capture plant’s operation impacts the capture solvent’s emissions. Surprisingly, this has been an open question since the process is so complex that it is not known how to set up corresponding process simulations. As in the case of the oxidation states, an inductive, data-driven approach is not constrained by this and, therefore, could give us insights into how the solvent emissions behave as a function of the operating conditions.

One underlying theme of the work presented in this thesis is that it is not computational chemists but their experimental colleagues that could benefit most from predictions enabled by machine-learning models. One fascinating development that might help in making machine learning more accessible are so-called foundation models. The closing chapter shows that such models can be fine-tuned with a few examples to give competitive performance across many chemistry and material science tasks.

However, most models are black boxes, and combining them with experienced chemists’ reasoning and even more background knowledge will likely yield the most progress. Combined with the progress thus far, this indicates that machine learning might have a larger impact on chemistry than in many other domains, such as computer vision.

ZUSAMMENFASSUNG

Die Entdeckung neuer Materialien ist essenziell, aber herausfordernd, und teuer. In vielen Fällen können Simulationen nützlich sein, um Material-Eigenschaften einzuschätzen. Für viele der interessantesten Eigenschaften sind Simulationen jedoch aufgrund prohibitiver Kosten oder weil es unbekannt ist, wie man eine geeignete Simulation aufsetzt, nicht durchführbar. Eine vielversprechende Alternative, um die Kosten der Vorhersage von Material-Eigenschaften zu reduzieren, oder überhaupt erst Vorhersagen zu ermöglichen, ist das datenbasierte Erlernen von Funktionen die Eigenschaften von Materialien oder Prozessen beschreiben. Obwohl die Durchführbarkeit dieses Ansatzes durch die Beobachtung gestützt wird, dass Chemiker aus Erfahrung lernen (Intuition), hat ein solch datenintensiver Ansatz einzigartige Herausforderungen. Erstens ist er auf geeignete Daten angewiesen, um die Forschung zu ermöglichen. Zweitens ist geeignete Infrastruktur erforderlich, um einen wissenschaftlichen Forschungsansatz zu ermöglichen, bei dem Forschungsergebnisse leicht verglichen und wiederverwendet werden können. Eine besondere dritte Herausforderung besteht darin, dass Kristallstrukturen oft zuerst in eine für maschinelles Lernen geeignete Form umgewandelt werden müssen (sogenannte Featurization).

Im ersten Teil dieser Thesis präsentieren wir Werkzeuge, die all diese Herausforderungen angehen. Wir erfassen Daten in einer maschinenlesbaren Form mit Open-Source elektronischen Laborjournalen. Anschließend verwenden wir Werkzeuge aus unserem "Ökosystem für retikuläre Chemie", das Datensätze, Werkzeuge für Datenaufteilung, Deskriptoren, und Benchmark-Hilfsfunktionen bereitstellt, um maschinell erlernte Modelle zu erstellen. Ein darauffolgendes Kapitel zeigt dann, wie solche Modelle verwendet werden können, um zu entscheiden, welches Experiment oder Simulation als nächstes durchgeführt werden sollte. Wir zeigen dies insbesondere im multikriteriellen Fall, welcher für die meisten Materialdesign-Probleme relevant ist.

Der zweite Teil dieser Arbeit verwendet Werkzeuge aus diesem Werkzeugkasten, um Probleme vom Atom- bis zum Pilotanlagen-Maßstab mit einem datengetriebenen Ansatz anzugehen. Auf atomarer Ebene zeigen wir, dass chemisch sinnvolle Deskriptoren verwendet werden können, um Oxidationszustände von Metallkationen vorherzusagen—eine Eigenschaft, die im Herzen der Chemie steht, für welche es aber keinen quantenmechanischen Operator gibt. Auf Pilotanlagen-Maßstab geht es darum, wie sich der Betrieb einer Kohlenstoffdioxid-Abscheidungsanlage auf die Emissionen des Absorptionsmittels auswirkt. Überraschenderweise war dies eine offene Frage, da der Prozess so komplex ist, dass nicht bekannt ist, wie entsprechende Prozesssimulationen aufgesetzt werden können. Wie im Fall der Oxidationszustände wird ein datengetriebener Ansatz durch dies nicht eingeschränkt und konnte uns Einblicke geben, wie sich die Emissionen des Absorptionsmittels in Abhängigkeit von den Betriebsbedingungen verhalten.

Ein zugrunde liegendes Motiv dieser Thesis ist, dass nicht die theoretischen Chemiker, sondern ihre experimentellen Kollegen am meisten von Vorhersagen profitieren könnten, die durch maschinelles Lernen ermöglicht werden. Eine faszinierende Entwicklung, die dazu beitragen könnte, maschinelles Lernen zugänglicher zu machen, sind sogenannte "foundation models". Im letzten Kapitel zeigen wir, dass diese Modelle mit wenigen Beispielen feinabgestimmt werden können, um eine kompetitive Leistung in vielen chemischen und materialwissenschaftlichen Fragestellungen zu erbringen.

Die Modelle sind jedoch Black Boxes, und wahrscheinlich kann der größte Fort-

schritt erzielt werden, wenn sie mit der Intuition erfahrener Chemiker und noch mehr Hintergrundwissen kombiniert werden. Kombiniert mit dem bisherigen Fortschritt, deutet dies darauf hin, dass maschinelles Lernen in der Chemie einen größeren Einfluss haben könnte als in vielen anderen Bereichen, wie zum Beispiel der Computervision.

ACKNOWLEDGMENTS

Given my background as a prior, it is unlikely that I ended up writing a Ph.D. thesis. In retrospect, a few individuals (*Rudolf (x2)*, and *Daniela*) during my time in high school were pivotal in encouraging me to go for a career in science, and I want to thank them for bringing me onto this path. Clearly, I also cannot thank my parents (*Danuta* and *Josef*) enough for supporting me on all the steps on this path.

Before I joined LSMO, I had the opportunity to do a lot of coding at BASF in the U.S. There, I was lucky to call *Brian* my colleague. Thanks for all the discussions back then and now and for the great collaboration—I'm looking forward to more to come!

My first steps in LSMO were under the tutelage of *Daniele*. Thank you for already during the Master Thesis giving me all imaginable freedom. That was the single best introduction to LSMO! Also, I could not imagine a more knowledgeable mentor with a profound knowledge of such diverse topics (from charge equilibration to trading).

Of course, a central part of my Ph.D. journey has been my advisor (not supervisor), *Berend*. My interactions with you, your leadership style, and how you think about science and the world have transformed me into a much more mature scientist (at least, I hope so; it is up to others to judge). I can still very vividly remember some of our first collaborative writing sessions in which I, as well as Mohamad, were just flabbergasted by your skills in scientific storytelling—we could have never envisioned presenting our work in such a way and were simply amazed that one could describe our work in such a way (however, I still need to go for the internship as a used-car salesman you recommended...). Another thing that makes a Ph.D. in your group unique is that you can get excited about (almost) any topic. You created a singular environment in which it is completely fine (and perhaps even expected) to work on graph neural networks, photocatalysis, biomass, software development, and carbon capture plants at the same time. Also, I can only thank you for involving me so early into grant writing and letting me teach classes—again leading to situations where Mohamad and I were staring at our inbox in disbelief that our names were between such world-renowned speakers (our first year of lectures at MolSim). Now, coming to the end of my Ph.D., I also want to thank those—Professors Natarajan, Gagliardi, Schier, and Züttel—who volunteered to take the time to participate in evaluating my work.

Doing the Ph.D. in a diverse environment like *Berend's* group gave me the opportunity to interact with many outstanding individuals—too many to list exhaustively in this text. Foremost, I want to thank my collaborators worldwide for introducing me to different aspects of chemistry and chemical engineering: From mixed-metal oxides at Heriot-Watt and Berkeley to carbon capture plants in the UK, Netherlands, and Germany. In particular, *Susana* and her team (including *Thea*) managed to keep us always organized. Other collaborations in my Ph.D. were much more self-organized and decentralized. Some, like the one with *Sterling*, simply emerged via contributions on GitHub. Others via shared Slack channels (*Aditi* and *Andrew*) and Twitter. Even though a large part of my Ph.D. time intersected with the COVID pandemic, I had the pleasure to also interact with many collaborators in the “real” world. *Mehrdad*, *Sauradeep*, and *Elias* always helped to keep the spirits high; *Sasha* and *Leo* were always open to chat about “niche” tech topics. And all the rest of the LSMO group—*Aisha*, *Anastasia*, *Balázs*, *Bea*, *Henglu*, *Maria*, *Miriam*, *Mish*, *Nency*, *Özge*, *Raffa*, and *Xiaoqi*—continued adding to this diversity, which *Evelyn* somehow managed to organize.

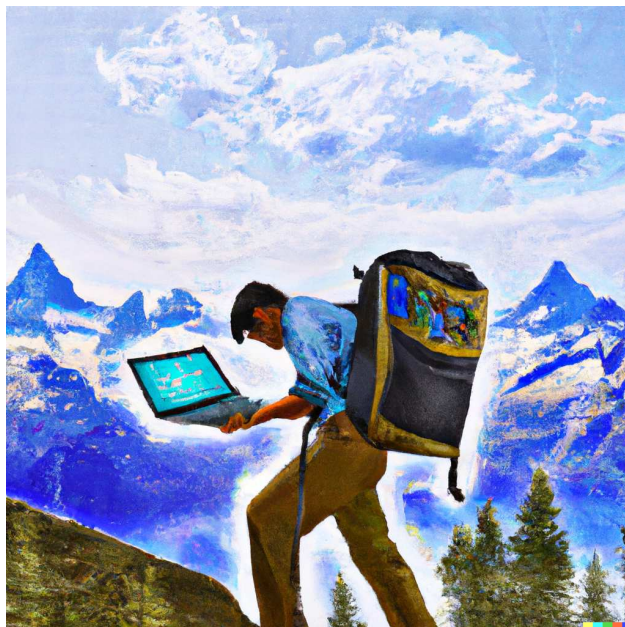


Figure 1: The impression the machine-learning model DALL-E¹ has of my Ph.D. time.

Often, there are roadblocks. Sometimes, those are because of a non-converging DFT simulation or due to a private issue—thank you, *Andres*, for always being available for both kinds of problems (until now) and always motivating me to push forward.

After around one year into my Ph.D. I met *Luc*, who is one of the most fascinating and clever people I have ever met, for the first time. Luc, thanks for all you taught me about software engineering and life—you make incredibly valuable contributions to our community.

The Ph.D. has not always been easy. However, I am so unfathomably lucky that you, *Tine*, decided to spend your life with such a strange geek. Thank you for reminding me of life outside the lab and for all our hikes and endeavors. Over the last seven (and counting!) years, you made me such a better person and gave me the best years of my life—I can't wait for the next steps of our journey. And, you know, *scripta manent!*

Vevey, April 25, 2023

CONTENTS

Abstract	3
Zusammenfassung	5
Acknowledgments	7
Introduction	11

i Data and infrastructure

1 Making the collective knowledge of chemistry open and machine actionable	21
2 An ecosystem for digital reticular chemistry	41
3 Bias free multiobjective active learning for materials design and discovery	67

ii Applications

4 Using collective knowledge to assign oxidation states	85
5 Machine learning for industrial processes: Forecasting amine emissions from a carbon capture plant	99

iii Discussion and outlook

6 Is GPT-3 all you need for low-data discovery in chemistry?	115
7 Conclusion and future research	131

iv Appendix

A Supporting Information for "Making the collective knowledge of chemistry open and machine actionable"	137
B Supporting Information for "An ecosystem for digital reticular chemistry"	147
C Supporting Information for "Bias free multiobjective active learning for materials design and discovery"	169
D Supporting Information for "Using collective knowledge to assign oxidation states "	199
E Supporting Information for "Machine learning for industrial processes: Forecasting amine emissions from a carbon capture plant"	257
F Supporting Information for "Is GPT-3 all you need for low-data discovery in chemistry?"	283
Curriculum Vitae	386
List of publications	389

INTRODUCTION

MATERIALS DESIGN AND DISCOVERY

The need for new materials

The progress of humanity is closely linked to materials. All the gains in quality of life over the history of our species would not have been imaginable without ever more complex use and development of materials.² Societies' consumption of materials skyrocketed with the conversion of chemical energy in fossil fuels into other forms, such as kinetic energy to move steam or Diesel engines. This development further accelerated with the generation of electricity and led to rapid increases in quality of life in certain parts of the world, which other parts of the world are yet to experience.^{2,3}

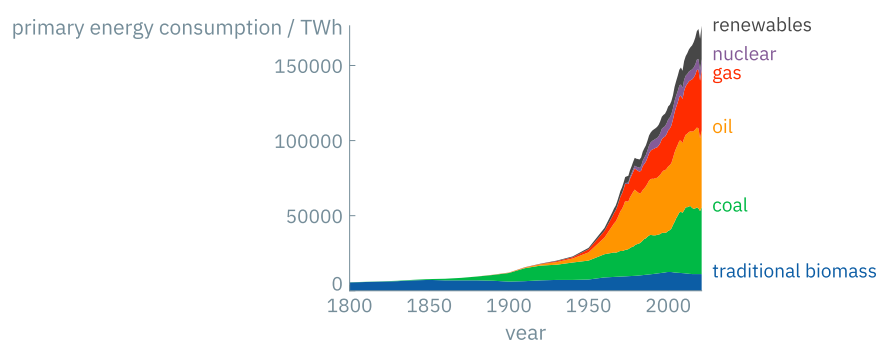


Figure 2: Global primary energy consumption by source. Replotted from Ritchie et al.⁴ Hydropower, wind, solar, modern biofuels, and other renewables are grouped into “renewables”. Traditional biomass refers to burning fuels such as wood or crop waste. The energy is given as substituted primary energy, i.e., accounting for inefficiencies in the conversion of fossil fuels. Ritchie et al.⁴ sourced the data from Smil⁵ and BP’s Statistical Review of World Energy.⁶

In 2019, the world consumed 176 431 TWh of energy (Figure 2).⁴ This corresponds to having more than 30 billion horses working for us all the time.* Since, however, the majority of this energy comes from the burning of fossil fuels (Figure 2), we emitted 33 622 Mt of CO₂. Over time this made the atmospheric concentration of CO₂ in the atmosphere rise from 278 ppm at the beginning of the industrial revolution to well over 400 ppm.⁹ This, along with increases in the concentration of other greenhouse gases, causes our climate to change.¹⁰ To tackle climate change, we need to decarbonize our society. However, progress in decarbonization has been slow. The analysis of Ritchie et al.⁴ shows that from 13.5 % low-carbon sources (renewable or nuclear) in 1994, we only moved to 15.7 % in 2019—while the consumption of fossil fuels continued to grow. Given the findings of the most recent climate models, this calls for rapid innovation.

Advances in materials can play a role in many different regards.⁸ To list just some:

* Or the equivalent of running nearly 60 billion graphics card with a peak power consumption of 350 W. The conversion into horses assumes a sustained power of ~ 0.912 mechanical horsepower per horse (even though the peak power can be 15.1 hp).⁷ Chu et al.⁸ estimated an equivalent of 25 billion horses for the energy consumption in 2012.

First, novel materials can help us generate electricity from renewable sources, for instance, by using more efficient and durable photovoltaics. Second, novel materials can help us convert energy more efficiently—for instance, by requiring less energy to light a screen or a room. Third, most renewable energy is intermittent. Innovations in materials can play a crucial role in developing materials for energy storage. Forth, novel catalysts or other materials, such as sorbents, can help make the chemical industry more sustainable. However, given the decarbonization trajectory thus far, we will need novel materials for a fifth application (which will be the focus of some parts of this thesis): To capture carbon dioxide.¹¹

Reticular chemistry

For all these applications, we need materials of often tremendous complexity. A commonly used approach to deal with complexity is to split a problem into different levels of hierarchy.¹² Reticular chemistry is an approach to chemistry that allows doing just that. Therefore, it is of particular interest for systematic studies of material design. The commonly used definition for reticular chemistry is the linking of *molecular building blocks* by *strong covalent bonds*. This implies a building block principle with which organic or metal-organic building blocks can be reticulated into potentially open framework structures—so-called, metal-organic frameworks (MOFs) and covalent-organic frameworks (COFs). It is, however, good to keep in mind that, in particular, the MOF label is often used quite inclusively, with some very prominent materials (or members of “MOF” databases) not being composed of molecular building blocks. The most crucial advantage of reticular chemistry is, however, that the last two decades have shown how reticular structures can be designed and manipulated on different length scales (Figure 3). These insights provide us with many degrees of freedom with which we can systematically design materials, particularly pore environments, for a given application. For these reasons, reticular materials will be a common case study in this thesis.

Challenges

The current time frame from material discovery to market is 15–20 years.^{19,20} Given the urgent need for new materials, this leaves considerable room for improvement. The time frame is so long because material discovery is challenging. Some of the challenges are the following.

COMBINATORIALLY LARGE SEARCH SPACE Chemists and material scientists are spoiled for choice.²¹ Some estimate the number of possible materials to be on the order of a googol (10^{100} , more than the number of atoms in the visible universe).²² In any case, chemical space is too large to enumerate and test by brute-force: If we assume a space of 100 trillion materials and “testing” one material (e.g., with molecular simulation) would take only 1 ms we would still need 3169 years to test all materials. This shows that brute-force testing of all possible materials is infeasible and indicates the need for accelerated approaches. However, foremost, it highlights a sampling challenge: We need efficient approaches to focus the search on the relevant parts of the design space as well as approaches that allow us to perform (surrogate) tests at low cost.

MULTIOBJECTIVE SEARCH The search in material space is further complicated because we need to consider more than one objective for any real-world application. In catalysis, for instance, we often want high selectivity and high activity, or in

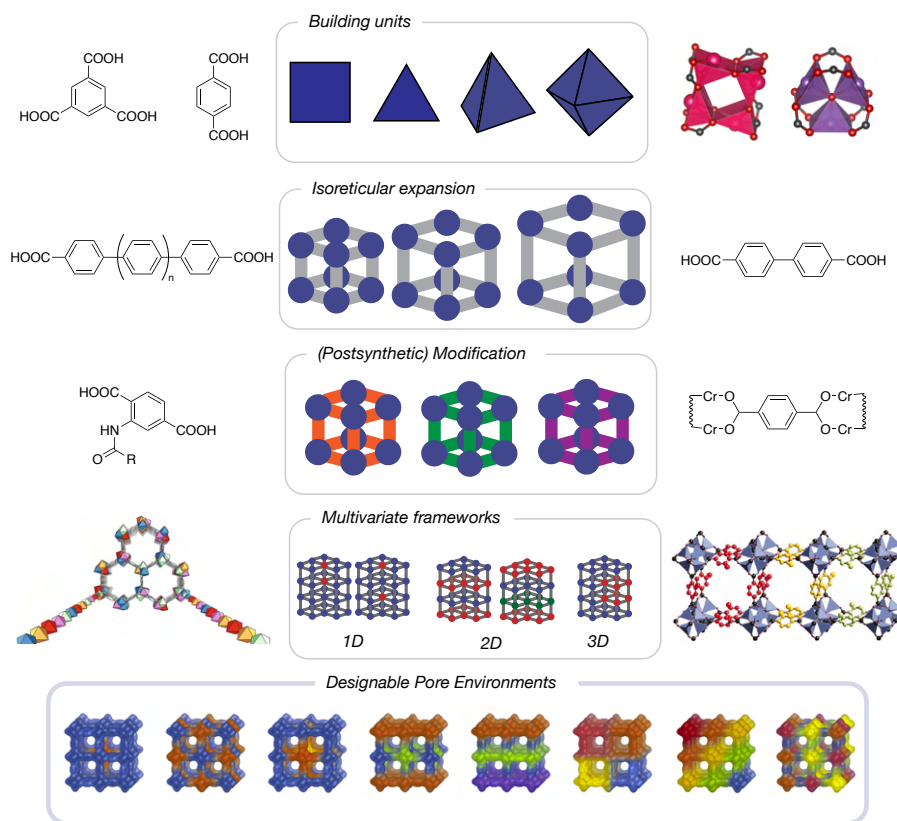


Figure 3: Reticular degrees of freedom across scales. At the most basic level, Reticular chemists can choose which building blocks they combine. The shape of the building blocks determines in which net they can self-assemble.¹³ Very early, it has shown that this allows for the design of specific properties of materials. For instance, using building blocks of the same connectivity, in which length is increased, one can tune the pore size (so-called isoreticular expansion).¹⁴ Some modifications might be hard to introduce before self-assembly. In this case, reticular chemists use post-synthetic modification to tune the pore environment without changing the building blocks.¹⁵ To design even more custom pore environments, reticular chemists can also combine different building blocks (e.g., different linkers with the same connectivity) to form a so-called multivariate framework. All these tools enable chemists to tune pore chemistry and shapes for given tasks at various steps in the synthesis. This illustration is based on a figure by Lyu et al.¹⁶ Some insets are based on illustrations from Deng et al.¹⁷ and Xu et al.¹⁸

metallurgy, we might want a strong and ductile material. Very often, the different objectives are actually competing with each other—for example, we often cannot increase activity without a drop in selectivity. The best we can do in such a case is to find the so-called Pareto frontier, which is the set of all solutions where we cannot improve an objective without making another one worse. While one might expect that this is a trivial extension of single-objective search, it is not. For example, one might consider converting a multiobjective optimization to a single-objective one using scalarization

$$\min_{x \in \mathcal{X}} \sum_{i=1}^k w_i f_i(x), \quad (1)$$

where w_i is the weight of objective function f_i for inputs x from the input space \mathcal{X} . The solution one obtains with this approach depends on the choice of the weights w_i . However, optimal choices for w are often unknown before the material design process is completed. This highlights the need for approaches to efficiently find the Pareto frontier without any particular choice of weighting (or other biasing in the search).

MULTISCALE SEARCH To find a material that can work in the real world, it is important to consider not only the atomistic or molecular scale. In many cases, descriptors (such as binding energies) on the atomistic scale do not correlate at all with the productivity of a material in an actual industrial process. A clear example of the importance of focusing on multiple scales at once is catalysis in MOFs. While the active site and the reaction mechanism might be very well described at the atomic scale, there will be no conversion if the reactant cannot diffuse to the active site (shape selectivity²³). Hence, a proper description needs also to consider diffusion pathways.

DATA-DRIVEN DESIGN AND DISCOVERY

Like many other fields, materials design went through different paradigms.^{24,25} For many centuries, the discovery and design of new materials has been dominated by empirical experiments. This has subsequently been refined with empirical laws or theories that eventually showed such an (emergent) complexity^[26] that they needed to be solved with computers. Thanks to the enormous growth in computational power and the refinement of computational approaches, so-called high-throughput virtual screenings have become indispensable tools in a material chemist's toolbox. For instance, it is now routine to screen hundreds of thousands of MOFs for carbon capture using grand canonical Monte Carlo (GCMC) simulations and classical force fields.²⁷ Similarly, researchers in surface catalysis made enormous progress by routinely computing adsorption energies of thousands of materials using density functional theory (DFT).²⁸

However, many phenomena are too complex to address with theory (alone). This might be due to system sizes or time scales that are too large to address with high accuracy (because, as Dirac said, the equations are “too complex to be solved”) or because we do not even know how to set up suitable simulations.

Interestingly, we have observational data from other, previous, or related instances in many such cases. For instance, we might have high-accuracy simulation data for small system sizes or experimental data for the yield of certain reactions or the stability of materials. Assuming that there is a relationship between structures and the properties of interest, we can use machine learning (ML) techniques to learn a

function $f(X) \rightarrow \hat{y}$ that takes some encoding of the structure, X , and returns an estimate \hat{y} of the property of interest, from data. Commonly, evaluating $f(X)$ is orders of magnitude cheaper than computing or measuring the property of interest y . This is one of the main promises of a data-driven approach: We can learn cheap high-fidelity approximations of an expensive-to-measure property y .

ML approaches in (material) chemistry have, in many respects, reached a considerable level of maturity. For instance, machine-learned models of potential energy surfaces (so-called ML force fields) have been scaled to describe hundreds of thousands of atoms at the accuracy of DFT with only a little higher cost than classical force fields.²⁹ Protein structure prediction using AlphaFold revolutionized how we research proteins.³⁰ In addition, tools such as Bayesian optimization are now common tools for optimizing reaction conditions.³¹

Challenges

Even though the progress has been impressive, there are still immense challenges ahead for data-driven (material) chemistry.

DATA — SIZE, BIAS, USABILITY First, a data-driven approach can only be used if enough data is usable—which means data that is well-documented, in a systematic format, and without too-much missing data, errors, bias,³² or noise. Much progress has been made in generating, curating, and sharing computational data. However, for some of the problems for which we cannot even perform simulations, the progress has been much more limited. For instance, it would be of enormous advantage if chemists could predict the optimal conditions for a given reaction. In theory, there is also a learnable correlation between the conditions and the reaction outcome—we can observe this with graduate students becoming more successful, i.e., experienced, with time. And in theory, we also produce a lot of data: tens of thousands of chemists are running multiple reactions every day—some of which work and many of which do not yield the expected outcome. However, this data is seldom shared, particularly of the “failed” reactions.³³ But precisely this data would be needed to develop tools that can provide chemists with a digital assistant that has learned from all the reactions performed worldwide. Addressing this requires—besides science policy and governance changes—better tools that avoid the systematic capture and dissemination of data becoming an afterthought. Importantly, this also involves tools that allow prioritizing what the most important next materials to investigate are.

TOOLING — STANDARDS, COMPARABILITY At the heart of the scientific method is the process of “standing on the shoulders of giants”. Science can only progress if we can compare and reuse each others’ work. Multiple recent studies highlight that the lack of thorough benchmarks, combined with troubling trends in research practice, might have led to an illusion of progress for certain applications of ML. In some cases, this is related to optimizing for metrics that are not relevant for downstream applications, whereas in other cases, there is data leakage, unrealistic performance measures, or no attempt to make the work comparable to prior attempts. These challenges, too, require, to some extent, efforts on the community level. However, they can also be bootstrapped by providing researchers with tools that facilitate the use of best practices and that make it straightforward to build on top of prior work.

REPRESENTATION — MULTISCALE, INVERTABILITY For use in machine learning models, molecules and materials must be converted into a suitable repre-

sentation. This is challenging because most ML approaches rely on linear algebra on matrices of fixed dimensions. Chemical compounds, however, come with vastly different numbers and types of atoms. In addition, many properties of chemical compounds do not change, or change in a particular way, upon application of symmetry operations such as translation or rotation. For crystalline materials, one additionally also needs to consider the periodicity. This has led to the development of a large variety of featurization approaches, many of which are the same idea expressed in a different basis set. To ensure progress in the field, it is important to harmonize and make them comparable. However, it is also important to move beyond the currently existing descriptors as almost none of them fulfills all the requirements (invertibility, permutation invariance, invariance/equivariance with Euclidean symmetry operations, smoothness). In addition, most existing approaches do not consider the multiscale nature, i.e., allow for a hierarchical representation of materials.

CONTRIBUTIONS OF THIS THESIS

This thesis summarizes some of our efforts toward data-driven material design. The thesis is split into two parts, where the first outlines tools developed to address the aforementioned challenges, which then find applications in the second part that showcases how data-driven approaches can find use from the atom scale up to the pilot-plant scale.

CHAPTER 1 As outlined above, machine-actionable data—including from “failed” attempts—is key for data-driven material design. This becomes already very clear when one attempts to build a ML model to predict the color of a MOF based on color names deposited in the Cambridge Structural Database (CSD); when doing so we realized that the noise in the data and the inappropriate representation is the main limitation for our modeling efforts.³⁴ These observations motivated us to improve the state-of-the-art in data capture in chemistry. A key element for this has been adapting an electronic lab notebook (ELN) for the challenges in a material science lab. The chapter outlines the vision we pursued when developing this infrastructure and gives an overview of our developments. These developments include, among others, the connection with simulation platforms (see [35]), the development of specific analysis tools, and the direct export of machine actionable data to repositories such as Zenodo (see [34] and [36] for examples). However, such an open-science infrastructure also has enormous potential for teaching by lowering the barrier to using computational tools and, in this way providing virtual laboratories (see Jablonka et al.³⁷).

Thanks to this infrastructure, multiple groups worldwide can routinely capture machine-actionable data.

CHAPTER 2 For data-driven material design, however, data alone is insufficient. Often, structures need to be encoded into fixed-size arrays to be acceptable inputs for commonly used machine-learning algorithms. Additionally, to be of any use, models must be evaluated in meaningful ways that are comparable with other works. This is seldom the case in the current practices for ML for MOFs. In particular, we show how models built based on data compiled from crystallographic databases are prone to data leakage—but we also provide methods for mitigating this problem. In fact, we report a complete ecosystem of tools that reticular chemists can use in every step of the machine learning workflow. We provide a new reference dataset and standard interfaces to commonly used ones, over 40 featurization approaches in a generalized implementation—including several novel ones, consistent model

evaluation tools, and a leaderboard. Overall, this ecosystem empowers novices and seasoned practitioners to use digital reticular chemistry best practices.

CHAPTER 3 Given a predictive ML model, we can use it to guide our search through chemical space. As outlined above, the search is particularly challenging if we consider more than one objective and do not know how to weigh the different objectives. To address this, we have introduced and implemented a multiobjective active learning approach that can recover the Pareto frontier efficiently without biasing the search. In collaboration with BASF, we applied this algorithm to the design of surfactants. Using mesoscale simulations, we could show how the active learning approach can enormously reduce the number of expensive simulations needed to recover the Pareto frontier—also in the case of missing data.

This provides us with a practical tool for finding the Pareto frontier in material design challenges and beyond.

CHAPTER 4 Armed with the tooling for building models, chapter 4 reports one case study in which ML could help solve an important question in material design. Oxidation states are a key element of chemical reasoning. They are such important for chemistry that they are even part of the names of chemicals. They are not only important for conceptualizing chemistry but also have very practical applications, for example, to initialize DFT simulations. However, oxidation states are not quantum mechanical observables and conventional approaches tend not to work on complex materials such as MOFs. By mining oxidation states from chemical names in the CSD and combining it with a chemistry-inspired featurization approach (using tools like the ones discussed in chapter 2), we could build a model that could assign oxidation states with high confidence. We could use feature importance analysis to reveal that this model reasons along chemically intuitive lines and highlights previous approaches' shortcomings.

This approach provides reticular chemists with a very practical tool for an important question (and is now also routinely used in our DFT workflows).

CHAPTER 5 Having visited the atom scale in chapter 4 (and already the mesoscale in chapter 3), chapter 5 showcases an example of how ML can be used on the pilot plant scale.

As discussed above, one of the ways material design can have a large impact is via carbon capture. The current state-of-the-art for industrial-scale carbon capture is amine scrubbing. Of course, one would like to avoid emissions of the solvent whose environmental impacts are not yet comprehensively understood. However, there is currently little knowledge of how the plant's operation impacts the amine emissions, and conventional process simulation approaches cannot be applied as all the governing mechanisms are not understood. Again, this is a problem of large complexity, which conventional techniques struggle to address. Using time series forecasting techniques, we could convert the data measured by our collaborators at a pilot plant, fed with a slipstream from a coal-fired powerplant, into a model that can forecast the emissions given the past and current state of the plant. We could then use this model to analyze experimental data and investigate potential operating conditions.

This work showcases how a data-driven approach can contribute to material science across scales: From the atom scale (Chapter 4) up to the plant scale (this chapter).

CHAPTER 6 The final chapter of this thesis explores a novel approach that can help close the loop to lab—which we discussed in the first chapter. A large challenge

with many of the ML tools developed in material science is that they are not easy to use (by domain experts such as bench chemists, who would often most profit from such tools). Additionally, it is often challenging to make reasonable predictions with little data. Interestingly, some of the most exciting results in ML have recently been obtained with very large models that some call “foundation models”. One interesting property of these models is that they tend to show emergent behavior, that is, the ability to solve tasks they have not explicitly been trained on. Inspired by this, we investigated if such models (like its archetype, generative pre-trained transformer model 3 (GPT-3)) can be fine-tuned for common low-data applications in chemistry. We found that even with elementary text representations, such as the name of a molecule, we can achieve performance that is competitive with or even outperforming baselines that had been fine-tuned for specific chemical applications.

This has potentially important applications for chemistry as a good baseline performance for practically relevant applications can be achieved with very little expertise and domain knowledge in machine learning.

Part I

DATA AND INFRASTRUCTURE

1

MAKING THE COLLECTIVE KNOWLEDGE OF CHEMISTRY OPEN AND MACHINE ACTIONABLE

MOF that can be made in one step in water x

About 3'430 results (0.97 seconds)

Synthesis Recipes

Nanocrystalline MIL-53
at room temperature
Díaz's MOF notebook
★★★★☆ 56 ratings
4 h at room temperature
Na₂BDC in H₂O added to
Al(NO₃)₃·9H₂O in water
breathing **abundant metal**

Nanocrystalline MIL-53
at room temperature
Díaz's MOF notebook
★★★★★ 89 ratings
72 h at room temperature
Na₂BDC in H₂O added to
Al(NO₃)₃·9H₂O in water
breathing **abundant metal**

MIL-808(Hf)
for methane storage
Dan's MOF blog
★★★★★ 19 ratings
12 h at 100 °C
HfCl₄ in water/formic acid
then BTC ligand
CH₄ storage

[show more results](#)

ABSTRACT Large amounts of data are generated in chemistry labs—nearly all instruments record data in a digital form, yet a significant proportion is also captured non-digitally and reported in ways non-accessible to humans and their computational agents. Chemical research is still largely centered around paper-based lab notebooks, and the publication of data is often more an afterthought than an integral part of the process. Here, we argue that a modular open-science platform for chemistry would benefit the entire chemistry community well beyond data-mining studies. Over the past few years, much progress has been made in developing technologies such as ELNs that aim to address data-management concerns. This is only one step towards making chemical data reusable, however. We highlight the importance of centering open-science initiatives around open, machine-actionable data and emphasize that most of the required technologies already exist—we only need to connect, polish, and embrace them.

CITATION This chapter is a preprint version of our perspective: Jablonka, K. M. et al. *Nat. Chem.* **2022**, *14*, 365–376.

CONTRIBUTION K.M.J wrote the article with editing by and contributions from B.S. and feedback from L.P. K.M.J also implemented several tools highlighted in the manuscript, including the pXRD, and isotherm analysis as well as the link to simulation platforms. K.M.J also contributed to the project as a member of the core development team of the cheminfo ELN.

1.1 INTRODUCTION

In the era when scientific results were published only on real paper, information compression was paramount. Due to limited page counts, most scientific data was never published. Now, we live in a digital era, and a large fraction of our data is captured in digital form. Yet, most scientific data that is collected is not published,³⁹ and the part that is being published is often in a form that makes it difficult for other researchers to build on top of it.

Scientists have also long been concerned about the reproducibility of results.^{40,41} In the face of this, most funding agencies insist on a commitment of the researchers to how the scientific data is managed (for instance, in the form of a data management plan, i.e., a clear outline of the types of data generated and used during a study, where and by whom they can be accessed, how and by who they are protected, how and by whom they can be shared or published), and often require to make all data publicly available. Having a data management plan is an important first step, but, as we argue here, it does not guarantee that the data will be shared in an easily findable, accessible, interoperable, and reusable (FAIR), and ultimately machine-actionable, form.⁴²

Recent advances in machine learning illustrate very clearly why chemistry would benefit from embracing open and reusable data. In chemistry, we have many problems of irreducible complexity,²⁶ such as the prediction of synthesizability, where the complexity arises due to the interaction of many diverse components (kinetics of side reactions, impurities, etc.), which are often not fully understood. Due to these unknowns and complex interactions, these problems seem impossible to address with the theory we have at the moment. However, to address those, data-intensive research might be the key. For example, many chemists would like a tool that recommends reaction conditions. One can envision building such a recommender system that harvests all the knowledge from all reactions that have been performed (including the “failed” ones) to recommend conditions for the desired reaction. However, building this tool will only be possible if all data is automatically collected in an interoperable and reusable form such that machines cannot only read the data but also autonomously discover relevant data sets and make decisions based on the collective of all data. This requires that machines cannot only parse the data but also understand it and its context, i.e., data must be machine-actionable.

Our key thesis is that if we want to advance chemistry with data-intensive research and address reproducibility problems, we must change how experimental data is collected and reported. Structured data is not enough; open data is also not enough. We need both (Thesis 1 in Figure 4) with additional tools, such as semantic web technologies, that allow chemists and their computational agents to understand the meaning and intent of the data objects.

To make this feasible, we envision a platform that seamlessly integrates the process of data collection, data processing, and data publication with minimal overhead for the researcher.

1. **Data collection** A key component of chemistry research is the collection of chemical data (for example, reaction conditions and characterization data). Ideally, the raw^{43–45} (characterization) data are directly captured from the instrument, directly converted into a standard structured form,⁴² in which all the important metadata are systematically added, and where the field names, such as “adsorption” or “pressure”, are linked to an open vocabulary (that defines the meaning of the terms and their relation).

One should not rely on individual chemists to manually perform file transfer, annotation, or conversion operations. It is not only time-consuming and error-

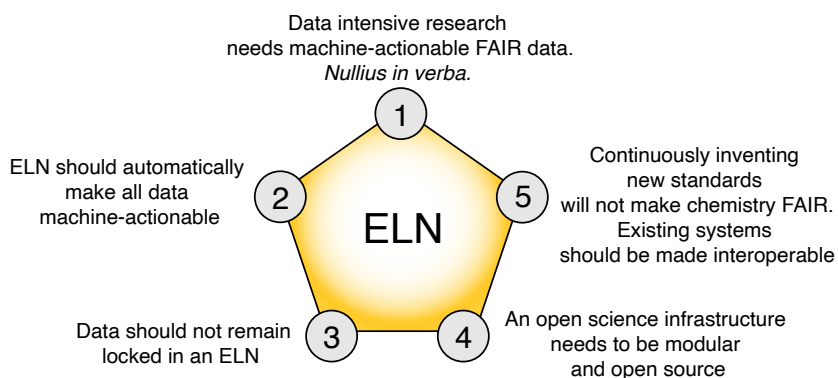


Figure 4: The five core theses of this perspective. Machine learning has fundamentally changed the way how data can be used in chemistry, and this, in turn, requires a change in how we report data. However, raw data is also needed to verify any conclusions presented in scientific principles—as stated with the “Nullius in verba principle” (take nobody’s word for it)—since the results presented in a paper are always a compression of the original research record.⁴³ For this, it is not enough if only a few groups create and share FAIR data; it needs to be embraced by all chemists. Importantly, this can only happen if there is little or no overhead of publishing all data in a FAIR, machine-actionable form. For this reason, the most important function an ELN can provide is to assist chemists in doing so; it is essential to avoid chemical data becoming an afterthought in the publication process. Following this logic, developers of ELNs need to work together towards this goal of machine-actionable open science. We can only expect this to be widely adopted if ELNs implement a common standard for data representation and exchange, also with computational tools,³⁵ and allow integration of reusable plugins which can be used to create a custom data management infrastructure that is interoperable with other solutions. Clearly, there will not be one perfect solution that works for all subfields of chemistry. However, we can start by reusing the many existing parts, making them interoperable and open-source the code, and in this way, create a practical solution that works today. This seems more effective than aiming for large-scale, all-encompassing, and over-complicated solutions. Importantly, developing new data formats will also not lead us toward the goal of FAIR chemical data.

prone, but more importantly, ensuring that all data are in a form ready for FAIR sharing should never become an afterthought but the very first step.

2. **Data processing and collaboration** Once we have converted our data to a standard form, we can apply the same analysis tools to all data types—making development dramatically more efficient. Research groups that use different instruments could compare the data directly and use the same analysis tools. Also, once all data are stored in a structured form, an ELN can make the data searchable. For example, if an instrument was incorrectly calibrated, the ELN could allow the users to search for all the spectra that were measured with a specific instrument configuration at a specific time range (or even automatically apply the correct calibration).
3. **Data publishing** Data that remains locked in an ELN is not useful for the community. As soon as the researcher is ready to publish a project, they could choose the relevant samples from the ELN and export them to a repository from where they can be used by machines, but also re-imported by other ELNs.

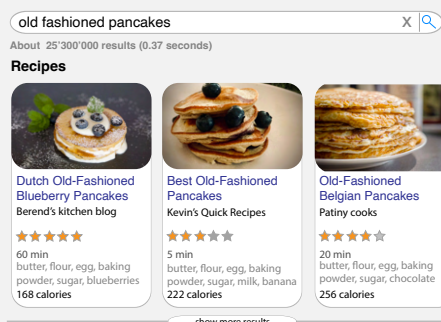
From this viewpoint, the ELN is the central hub for all chemical research, where analyses can be requested, analyzed, shared, published, and integrated with other

platforms—and also, a place to take notes. But we want to emphasize that the most important functionality an ELN can provide is to automatically convert the data into an open, standardized, and interoperable form (Thesis 2 in Figure 4). Only in this way can we leverage web technologies that can allow computational tools to autonomously understand data and hence provide more meaningful (search) results (see Box 1). Note that this differs from the functionality most current ELNs offer. Please note that the majority of current ELNs only store data digitally as an attachment, but they do not convert it into such a reusable form (Thesis 2 in Figure 4).

Box 1: Machine-actionable data in chemistry

Data structured in standardized ways can make information findable and interpretable by chemists and their machines and thus can enable humans, as well as their computational agents, to perform actions based on the interpretation of the data.


If we perform web searches, major search engines display meaningful information (sometimes even formatted in infoboxes with tables that allow for easy comparison) and can show related content instead of just a list of hyperlinks. For instance, search engines will show, when queried for “old fashioned pancakes”, a compilation of recipes from different sites—similar to what is shown in our example (see the right panel in the figure). This is possible because the websites embed the information in a standardized form, as shown here, into the website using in-page markup, typically `schema.org` (as in the code snippet on the right-hand side of our example). In summary, the recipe data is reported in a standard, open format, using linked vocabularies, described with metadata, and accessible under uniform resource identifiers (URIs).



old fashioned pancakes

About 25'300'000 results (0.37 seconds)


Recipes



Dutch Old-Fashioned Blueberry Pancakes
Berend's kitchen blog

★★★★★


60 min
butter, flour, egg, baking powder, sugar, blueberries
168 calories



Best Old-Fashioned Pancakes
Kevin's Quick Recipes

★★★★★

5 min
butter, flour, egg, baking powder, sugar, milk, banana
222 calories



Old-Fashioned Belgian Pancakes
Patinny cooks

★★★★★

20 min
butter, flour, egg, baking powder, sugar, chocolate
256 calories

[show more results](#)

```
{
  "@context": "http://schema.org",
  "name": "Best Old-Fashioned Pancakes",
  "datePublished": "2000-00-01T21:53:33.000Z",
  "recipeIngredient": [
    "1 cup flour",
    "3 teaspoons baking powder",
    "2 tablespoon white sugar",
    "1 cup milk",
    "1 banana",
    "2 egg",
    "2 tablespoons melted butter"
  ],
  "nutrition": {
    "@type": "NutritionInformation",
    "calories": "222 calories"
  },
  "aggregateRating": {
    "@type": "AggregateRating",
    "ratingValue": 3,
    "ratingCount": 179,
    "itemReviewed": "Best Old-Fashioned Pancakes",
    "bestRating": "5",
    "worstRating": "1"
  }
}
```

Similar markup has been used to encode the COVID-19 announcements, such as special opening hours or prevention measures, on websites (including those from the US federal government⁴⁶) which search engines could then highlight.⁴⁷ Readers can find such markup by using the “inspect” or “view page source” tools of their browser (which can typically be accessed with a right-click on the page) and then searching for “schema.org”.

If similar metadata were embedded in, for example, all published spectra (for example, NMR, IR, Raman, XPS), we could simply use a web search to find all spectra published for a particular chemical in a particular time period. With proper semantic annotation, we could, for instance, also specifically query for “vibrational spectroscopy” to receive IR, Raman, and sum frequency generation spectra. Clearly, one can also envision using such standardized structured data for synthesis “recipes”. This might facilitate the comparison of different synthetic conditions and also incorporate the feedback of other chemists. The bioschemas⁴⁸ and Materials Schema efforts

attempt to move the life and materials sciences closer to this ideal.

Some concrete steps and questions chemists can ask themselves to check their data objects for reusability and reliability,⁴⁹ are the following.

- **Data should be structured using standard, open conventions: Can others (humans and machines) easily use my data objects with their tools?**

This ensures that others can read the data. In practice, this means that an open format is always preferred over a proprietary one. Standard formats (JavaScript object notation (JSON), XML, JCAMP-DX) ensure that others can use standard tools to read the data objects.

- **Entries in a data object should use a controlled vocabulary and ideally reference an ontology: Can others, humans and machines, easily understand the meaning and format of all fields in the data object?**

ontologies explain the meaning and relation of the fields. For example, when reporting a band gap, one must ensure that the field “band gap” can be correctly interpreted (as it might refer to the optical, fundamental, or transport gap). A key challenge is that the documentation for the dataset is often transported “out of band” if the data, e.g., in the Supporting Information of a paper, instead of directly “in band” with the data object. JSON-LD⁵⁰ (see Figure 6) and CSV-LD⁵¹ are great ways of providing the context “in band” with the data.

- **Data should be annotated with metadata, ideally indicating the provenance of the data: Do others, humans and machines, understand where the data comes from and the context within which it was produced?**

This information can, for example, be important when issues with the data arise. For example, metadata might help us find that all reactions were unsuccessful because a batch of the (commercial) starting material was impure or the humidity or temperature in the room was too high. In chemistry, there is no widely used standard for recording basic metadata of ELN entries, even though proposals like the elnItemManifest, which builds on the Dublin Core scheme, have been made.⁵²

- **Data should also be uniquely identifiable, and citable, using a stable, and indexed, URI: Can others, machines, and humans, rely on finding the data in a stable form, and see the history, and do they know the usage conditions?**

If data is aimed to be reused, it should be accompanied by a license that allows this (for example, a creative commons license such as CC0, a donation to the public domain, or CC-BY, which also requires attribution of the originator). Using a URL that points to a GitHub repository or personal web page is hereby not enough—the problem is that the content of such URLs can easily change, for example, by deleting a repository on GitHub (a phenomenon called link rot). For this reason, data should be shared via data repositories where it is assigned a stable identifier (such as a DOI) that is guaranteed to point to the content. Also, repositories will ensure that the metadata and identifier are indexed and can be found. For organic chemistry, a domain-specific repository is the chemotion repository.⁵³ Also, for identifiers (for example, for samples and instruments), it is best to use hypertext

URLs such that others, humans and machines, can easily look up those identifiers.

Additionally, others should be able to find out the history of changes in the data and if it is still maintained. Most repositories can provide this functionality as “versions” of the dataset.

- **Data should be linked to other data: Can others, humans and machines, easily find related data (e.g., computational data supporting experimental work)?**

Linking data provides context and lets users of the data discover related datasets. From our recipe example, we can imagine that related content can give us useful information, for example, direct us to the recipe the original author was inspired by. In the chemistry context, we should link, for instance, to computational work supporting experimental measurement or to crystal structures deposited in another database.

Over time, an “insane”⁵⁴ number of different ELNs and laboratory infrastructure management system (LIMS) have been developed. Many of these different ELNs have been compared in previous works (for example, by the Harvard Medical school, the Library of the University of Cambridge, LIMSWiki, or peer-reviewed articles^{55–58}). In this perspective, we aim to focus on the ideas and design principles that we think are essential for creating a successful open-science infrastructure—for the full lifetime of data from inception, creation, and processing to publication. Since the infrastructure we propose to embrace is already implemented in parts, we will review some examples (from Table 1) that we think to offer some key aspects of such an infrastructure that supports open science. Similarly, we will highlight examples where chemical data has already been shared in a reusable form. Taking into account the many attempts to generate new data schema—describing the abstract structure of the data—and file formats for chemical data, we propose that a more efficient route to open science would be for the chemistry community to embrace and connect existing systems instead (Thesis 4 in Figure 4).

1.2 DATA CAPTURE, DATA PROCESSING, AND DATA PUBLICATION

To be practical, the data capture step needs to be as close as possible to the way chemists work while ensuring that the chemical data they generate can be practically reused by other researchers. We give examples of what “machine-actionable data” means in Box 1.

In chemistry, most samples in the lab are produced with a chemical reaction. Trying to predict the conditions at which a reaction optimally can take place is still one of the major challenges in chemistry. Machine-learning methods are expected to help us in this area.⁶⁹ However, for this to work, we need to report data in a format that can be used in machine learning and also report “failed” experiments.^{33,70} One can easily see the dilemma here; if an experiment—after 99 “failed” attempts—finally works, there is little motivation, if any, for a researcher to spend 1 % of their time in reporting the one successful experiment and the remaining 99 % of the time on the “failed” ones.

Table 1: Examples for some LIMS/ELN systems. Note that we only list open source solutions as we believe that successful solutions must be developed from reusable building blocks given that the requirements for data management in chemistry are so diverse.

system	key feature
Chemotion ELN ⁵⁹	chemistry centered user interface, integration with some databases like SciFinder, can perform basic sanity checks/quality control, for example, checking peak assignments using simulations ⁶⁰ —i.e., small tools that simplify the life of chemists, tightly integrated with the chemotion repository. ⁵³
openBIS ⁶¹	modularity via plugins, integration of Jupyter notebooks (computational environments that allow for literate programming, i.e., the combination of text and visualization with code, that have become a standard across sciences) for custom data analysis. Can be used as metadata repository for large files that can be linked and stored in other locations.
cheminfo ELN ⁶²	large ecosystem of data analysis and conversion packages centered around one common data object, modular architecture. FAIR data is the center of all operations, a chemistry-centered interface.
LabTrove ⁶³	the ELN can be a form of a blog which allows for open notebook science (i.e., making the full research record openly available on the web)—as popularized via “Open Source Malaria” ^{64,65} —which highlights the social components of research and allows for new forms of collaboration.
eLabFTW ⁶⁶	trusted timestamps that could be used as legal “proof of discovery” to defend a patent.
Sciformation ELN ⁶⁷	integration of chemical libraries (for instance, to fill in basic data such as molar masses) and support for analytical requests to a central service, definition of workflows (for instance, for the sequence of steps for sample preparation), audit trail functionality. The successor of the open eventory.
Kadi4Mat ⁶⁸	Integrates (to some extent) a data repository with an ELN, with flexible user-definable metadata schema. Allows defining workflows that perform a sequence of tasks, such as extraction and processing, on the data.

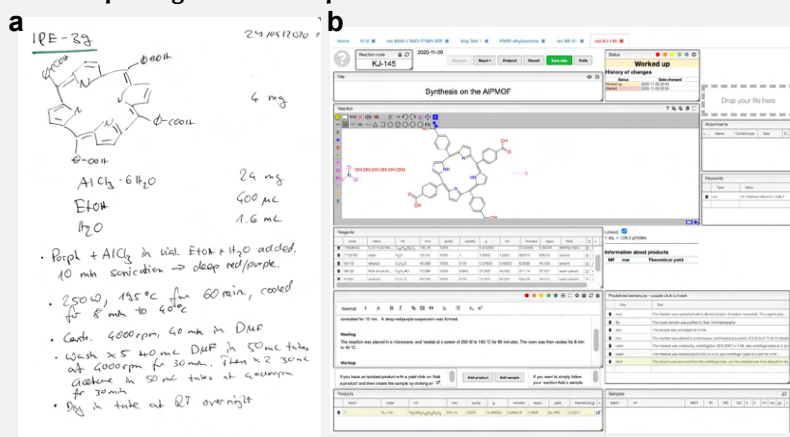
CAPTURING SYNTHETIC DATA In chemistry, the number of possible steps and combinations of steps is nearly infinite. For example, the order in which the reagents are added can decide whether a reaction will be successful or not^{71,72}—and any machine-learning efforts will fail if such information is not reported correctly. This is exactly what is missing in many of the existing databases. For example, by mining the patent literature⁷³ one can obtain a wealth of information on which chemicals can be synthesized.⁷⁴ However, the actual procedure of the syntheses cannot be mined systematically: The order of addition, the heating, the stirring, and, of course, the workup and purification. And the situation is even direr for inorganic chemistry.⁷⁵ Similarly, all databases contain no information about the attempts that did not work and are biased toward certain reaction types.^{76–78} This lack of reports on “failed” reactions adds to other factors that lead to certain types of reactions being more prominent than others—for example, looking into the most used reactions in medicinal chemistry, Brown found that amide formation was mentioned at least once in about half of the selected set of manuscripts, published in the *Journal of Medicinal Chemistry* in 2014 (ref. [79]).

Ideally, capturing synthesis information is finding a balance between the flexibility of the sheet of paper, on which chemists can record anything they want in any format they like,⁸⁰ and imposing a structure that can be easily reused for machine-learning applications. The flexibility is key to ensure chemists widely adopt the tool,^{55,81} while from a data management perspective, a highly structured database (for example, filled via a long form) would be much easier to use. In high-throughput experimentation settings, the latter might be a natural approach, but for many manually created, small data sets,³⁹ this might not be a feasible approach as capturing all possible scenarios would result in such a gigantic form that chemists would need special training to navigate it.

Among the different ELNs no consensus has been reached about this design point.

Some allow complete flexibility and have a look-and-feel of a typical note-taking app. Therefore, one would need natural language processing to make the information machine-readable, which unavoidably leads to information loss. On the other end of the spectrum are those with a lot of structure with designing a new form for every eventuality, which might be ideal for machine learning but poses a burden for non-routine chemistry. A possible solution for these challenges, and which is implemented in the chemotion and the cheminfo ELNs, is to stick to the text-based form chemists are used to but to combine it with templates to structure the text. This hybrid approach is described in Box 2. In practice, we found that some free text fields are always required to give chemists the flexibility to express their motivation, thought process, and interpretation. This can be captured via specific fields, such as related literature or spectral annotations. For many other parts, the free, potentially unstructured thought process is exactly what one would like to capture (for example, to annotate when an experiment failed for an unexpected reason, such as a beam drop at the synchrotron).

Box 2: Capturing the reaction process



A paper notebook (as the one in the left image, **a**) would typically read

...we added 10 mg of chemical A (batch 4, see page 25), 5 mg of chemical B (batch 5 see pg 61 of notebook 6 of Colleague Y), 5 mg of chemical C (Chem-R-Us) in a 50 %DMF/50 %water mixture and put the solution in the oven Y for 11 h at 70 °C ...

One can envision that this is a simple step in a complex synthesis in which we are trying to find the optimal conditions for a particular reaction. The question is now how to convert such chemical data into a format that can be practically mined and possibly used for machine-learning studies and yet maintain a level of flexibility that is essential for chemists.

The idea of such a workflow is to find a compromise between being able to easily extract process variables (like the heating time and temperature) while still providing the chemist with a natural interface of a text and structure editor such that the structure of the ELN remains similar to what they are used to from paper-based notebooks (panel **b** in the figure). In this scenario, research groups—or ideally, consortia of research groups) can define predefined sentences (with fillable fields) for common operations like heating to reflux, filtering that can be inserted with a shortcut such that the outcome is

...we added **R1** (xR1 g), **R2** (xR2 g), **R3** (xR3 g), in a **Y**%**R4**/(100-**Y**)%**R5** mixture and put the solution in the oven **y** for **t** h at **T** °C

...

in which all bold elements resolve to some **URI**. If behind the scenes the predefined sentences map to a well-defined set of concepts (in standard vocabularies), the description also becomes independent of the language it is written in.

The real advantages of this approach become clear if we look at the different shortcuts. Each reagent (which can be a previously produced sample or one from a manufacturer catalog) can be referred to via the hyperlink. Following those links, the researcher has direct access to all information about the provenance of the reactants and, from the order of the links, can extract the order of the synthesis procedure as it is typically described sequentially. At the same time, this approach reduces the time needed to record experiments as most of the usual operations can be inserted with tab completion, and observations such as “the solution turned blue” can be seamlessly integrated. In this context, it is important to realize that the ways in which observations are typically reported in chemistry are inadequate.³⁴ For example, colors are usually reported as color names (such as “dull blue”) in papers and databases, which are subject to perceptive spread and which, therefore, can limit the utility of such observations for replication studies or machine learning approaches. In the case of colors, for example, we recommend recording images with color calibration cards, from which a numerical color value can be easily extracted. At the same time, the image also gives information about the material’s morphology.

Another promising approach is lab automation, as proposed by the company labforward, that, for example, allows to connect balances, rotary evaporators, or vacuum pumps to an ELN and in this way, capture (automatically) more data in a structured and objective way.⁸²

DATA FORMATS AND METADATA After a sample has been synthesized, it needs to be characterized. Hereby we want to ensure that researchers all over the world, as well as their computational agents, can use the data. Data models, which describe how data is stored in a data format, and metadata (that describe datasets) are not the typical focus of a chemist. However, a lot of chemical data is currently stored in a wide variety of proprietary files (see Table 7). In the short term, this might not look like a real problem, but in the long term, this is not sustainable. For example, one can lose access to all files once the software license associated with a particular piece of equipment expires. Or, collaborators in another institute that want to use the data do not have access to the same software. And a hodgepodge of inconsistent formats hampers data mining efforts.

Requiring all individual researchers to manually convert all their spectra to a standard format will be a large, potentially insurmountable, and non-scalable burden on the researchers. Therefore, an essential step in progressing towards such an open platform is to convert the data to a standardized structured form before it even enters the ELN (Thesis 2 in Figure 4). This is an essential service an ELN must provide to a user. That is, the ELN will take the data as it is provided by the spectrometer and convert it into a standardized form. The cheminfo implementation, for example, uses JCAMP-DX files (Joint Committee on Atomic and Molecular Physical Data Exchange format, see Figure 5 for an example) as standard representation for most spectra. This format has been recommended by IUPAC for many spectra together with recommended vocabulary,⁸³ and is also recommended by the chemotion ELN, and used in the Open Spectral Database.⁸⁴

<pre>##TITLE=340343 V123413/A1A ##JCAMP-DX=4.24 ##DATA TYPE=NMR SPECTRUM ##DATA CLASS=XYDATA ##ORIGIN=agfavnmr ##OWNER=Jon Doe ##SPECTROMETER/DATA SYSTEM=Varian GEMINI 2000 300 \$\$ Varian Associates, Inc., VNMR Software \$\$ VNMR Version 6.1 Revision B, December 4, 1998 \$\$ Tue Jan 13 15:28:35 WET DST 2004 ## OBSERVE FREQUENCY= 299.9328561 ## OBSERVE NUCLEUS=1H ## FIELD=7.04 \$\$ Tesla ## ACQUISITION TIME=4.9996040 \$\$ seconds ## AVERAGES=64 \$\$ number of transients ##\$REFERENCE_POINT=224.207 \$\$Referencing label ##DELTA=-0.001027655 ##XUNITS=ppm ##YUNITS=ARBITRARY UNITS \$\$ mm on paper ##XFACTOR=0.001027655 ##YFACTOR=160.199999131 ##FIRSTX=16.089531599 ##LASTX=-0.746542441 ##MAXY=127.420862 ##MINY=-47.142647 ##NPOINTS=16385.000000000 ##FIRSTY=-0.015555 ##XYDATA= (X++(Y..Y)) 0 -0.000111519 0.000307130 0.000000878 0.000275188 0.000287949 5 -0.000106869 0.000208266 0.000100644 0.000211087 -0.000004851 10 -0.000027479 -0.000023142 0.000092236 0.000141324 -0.000004786 15 0.000216486 -0.000011140 0.000437822 0.000111168 0.000372978 20 -0.000003241 0.000114166 0.000219968 0.000067024 0.000385761 25 -0.000325340 0.000099498 -0.000343668 -0.000042190 0.000055055 30 0.000217357 0.000106220 0.000075300 0.000138421 0.000490292 ##End=</pre>	<p>header: provides core and additional metadata as labeled data records</p> <p>some core metadata elements (title, JCAMP-DX version, data type, origin, owner) must always be provided</p> <p>comments can be inserted using \$\$</p> <p>for many spectrum types, such as NMR, specific fields (e.g., OBSERVE NUCLEUS) were defined by the IUPAC working groups</p> <p>private (user defined) labels can be added using ##\$</p> <p>indicates start of data in XYDATA format the actual data in form</p> <p>$X_1, Y_1, X_2, Y_2, X_3, Y_3, X_4, Y_4, \dots, X_N, Y_N$</p> <p>indicates the end of the file</p>
---	--

Figure 5: Fragment of an NMR spectrum serialized to a classic standard format. This is an example of a JCAMP-DX file. This format is a widely used International Union of Pure and Applied Chemistry (IUPAC)-recommended format for spectra that is, for example, supported by the cheminfo and chemotion ELNs. Also, spectra in many databases such as the NIST webbook or the Infrared & Raman Users Group (IRUG) Spectral Database can be downloaded in JCAMP-DX format. A JCAMP-DX file can contain multiple blocks of labeled data records (LDRs). That is, one can store multiple related spectra (such as repeated measurements) in the same file. All data blocks must contain a CORE header with basic metadata such as OWNER, DATATYPE. The IUPAC working group also provides a vocabulary of further global labels, such as for the temperature/pressure/CAS-number. Data can also be compressed using various compression schemes. Note that the JCAMP-DX format is only one old standard, and many others have been proposed. The JCAMP-DX format, however, does allow for the addition of an unlimited number of private labels by using the ##\$ prefix, which allows every system to tailor the format to its own needs. Drawbacks of this format are, however, that it does not come with native, standardized support for semantic web features (such as linking to a vocabulary) and, in contrast to formats like XML, CSV, or JSON, that it is not natively supported by many general-purpose tools.

<pre> { "@context": ["https://stuchalk.github.io/scidata/contexts/scidata.jsonld", { "sdo": "https://stuchalk.github.io/scidata/ontology/scidata.owl#", "cao": "https://stuchalk.github.io/scidata/ontology/cao.owl#", "qudt": "http://qudt.org/vocab/unit/", "obo": "http://purl.obolibrary.org/obo/" }], "@base": "https://mysite/nmr/scidata/" }, </pre>	<p>provides prefix (i.e., shorthand) for and reference to vocabularies used in this file</p>
<pre> "@id": "https://mysite/nmr/scidata", "@graph": { "@id": "https://mysite/nmr/scidata", "@type": "sdo:scidataFramework", "scidata": { "@id": "scidata/", "@type": "sdo:scientificData", "methodology": { </pre>	<p>provides root address under which this resource can be found</p>
<pre> "@id": "methodology/", "@type": "sdo:methodology", "evaluation": ["experimental"], "aspects": [</pre>	<p>the measurement parameters/methodology relative address of the methodology part (relative to the root address)</p>
<pre> { "@id": "measurement/1/", "@type": "cao:CAO_000152", "technique": "obo:CHMO_0000591", </pre>	<p>technique (NMR) described using chemical methods ontology (CHMO)</p>
<pre> "settings": [{ "@id": "setting/1/", "quantity": "frequency", "property": "Observe Frequency", "value": { "@id": "setting/1/value/", "@type": "sdo:value", "number": "300.03180", "unitref": "qudt:MegaHZ" } }] </pre>	<p>units defined using the QUDT vocabulary</p>
<pre> ... "dataset": { "@id": "dataset/", "@type": "sdo:dataset", "source": "measurement/1/", "scope": "substance/1/", "dataseries": [</pre>	<p>the actual datasets (the free induction decay)</p>
<pre> { "@id": "dataseries/1/", "@type": "sdo:independent", "label": "Excitation frequency (Hz)", "axis": "x-axis", "parameter": { "@id": "dataseries/1/parameter/", "@type": "sdo:parameter", "quantity": "frequency", "property": "Radiofrequency", "valuearray": { </pre>	<p>reference to the chemical defined at another point in this file</p>
<pre> "@id": "dataseries/1/parameter/valuearray/", "@type": "sdo:valuearray", "datatype": "decimal", "numberarray": [4184, -617.85094858], "unitref": "qudt:HZ" } }] </pre>	<p>the valuearray type describes a list of doubles</p>
<pre> } } } } } } </pre>	<p>the datapoints (shortened for this figure)</p>

Figure 6: Fragment of an NMR spectrum serialized to a modern standard format. We show another NMR dataset (taken from the SciData website from the Chalk Group at the University of North Florida) serialized to JSON-LD using the SciData data model.⁸⁴ One important part of the JSON-LD file is the `@context` field. The values in this field link to the vocabularies used for naming things in this data file. For instance, for units, the vocabularies provided by qudt are used, whereas the method is described using the chemical methods ontology (from which it is clear that, for instance, NMR spectroscopy is—similar to electron spin resonance spectroscopy—a magnetic resonance method). Almost all modern programming languages support reading such JSON files. The `@type` field can describe the format of the data, for instance, to let a computer know that it can expect a list of doubles. Different parts of the file (such as methodology and the dataset) can be accessed by their own address.

But, in principle, any other format (see Table 9) can be used as long as it is standardized and openly documented. Indeed, some newer formats have native support for advanced features such as linking to standardized vocabularies and might be preferable (see Figure 6 for an example). For example, there have been efforts (spearheaded by the pharmaceutical industry) to develop a “Unified Data Model” for compound synthesis and testing, or the “Allotrope Data Format” that tries to collect the full data life-cycle in one file. Some, like the auto protocol or XDL,⁸⁵ even try to capture the link between hardware (like reaction vessels) and the synthesis steps in a way that can be both understood (and executed) by robots and humans.

One can argue that some existing formats and data schema are old-fashioned and that one should develop new ones. However, anyone proposing a new format should realize that if a characterization method has N formats provided by the instrument manufacturers and M “standard” formats are invented, we need to write and maintain $N \times M$ conversion programs and M^2 programs to be able to compare the different “standard” formats. This indicates that updating existing solutions and making them interoperable can be more productive than creating new ones (Thesis 5 in Figure 4).

It is important to note that data becomes much more useful and interoperable if it is linked and described using a controlled, hierarchical vocabulary, i.e., an ontology. A formal ontology would allow us to infer information from the context encoded in the vocabulary. For example, we might have Raman and infrared spectra and the cities of the measurement stored in our database. The ontology will not only remove ambiguities in the cities’ spelling but also tell us which cities to include if we search for, say, all organic samples with vibrational spectra measured in a particular country. At the technical level, this is enabled because the ontology will encode that both infrared and Raman spectroscopy are forms of vibrational spectroscopy and that cities are located in countries. That is, it allows us to go from machine-readable to machine interpretable on a global scale (global because the terms are standardized and shared via URIs). In practice, however, ontologies (and related semantic web technologies) remain underused. The main reasons for that are likely that the diversity of ontologies might be too high and that existing ones are poorly integrated.⁸⁶ Clearly, we cannot expect chemists to manually annotate their data using an ontology. An ELN needs to do this automatically in the background. However, for this to be practical, ELN developers need to connect with other initiatives to register, standardize, link,⁸⁷ and adopt ontologies.

Let us now assume the ideal and most chemists have settled on a standard data reporting form (for the most important characterization techniques in a sub-domain, such as gas adsorption isotherms, X-ray adsorption spectroscopy, cyclic voltammetry), and we also accept that open science should never be an afterthought. This implies that the ELN must take the file in whatever form it comes from the instrument, convert it into this standard form, and permanently connect it to the chemical that was characterized (Figure 7). Such conversion tools (see Table 7 for examples) can be developed independently of each other and reused in all ELNs. For instance, the chemotion ELN reuses some of the libraries that we have been developing for the cheminfo ELN (cheminfo.github.io). Such common conversion tools would also incentivize adopting a common schema.

PROVENANCE OF DATA One crucial step in this process is to match the spectrum to the correct sample. A URI system (can be printed as barcodes) can help to avoid mistakes in this step. For instance, in the cheminfo ELN, scanning the barcode will create the upload information for automatic importation from the computers to which the spectrometers are connected. From there, the system can take the file from the computer, convert it to the standard form, and store it as an attachment to a

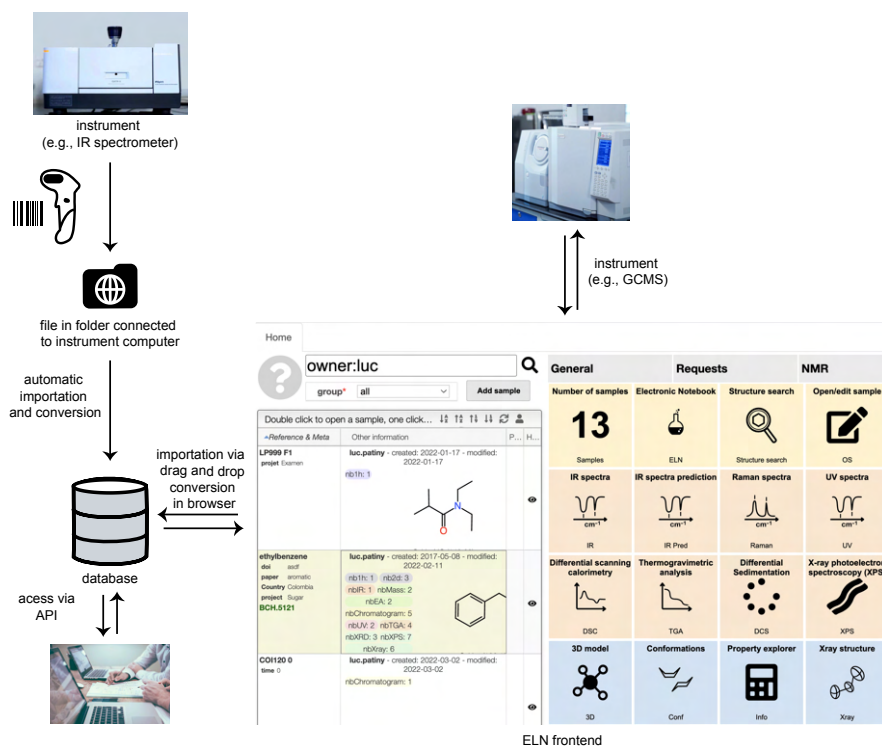


Figure 7: Overview of a possible importation procedure of the ELN. If an instrument is coupled to the network, one can upload the analysis result directly into a database by scanning the barcode on the sample. Alternatively, one can upload files via drag and drop through a web interface (frontend). In both cases, the ELN ensures that the data are converted to a standard form such that anyone with a web browser can visualize and further analyze the data. Other parties can access, for example, using an access token mechanism,³⁵ the data using a representational state transfer (REST)-application programming interface (API) or the published data on a repository. Importantly, all steps can take place from a different location, hence enabling collaboration. This data infrastructure is implemented in the open-source chem-info ELN.

sample that has been created in the ELN (for example, as a product of some reaction). This automatic importation not only makes it much easier and less error-prone for the chemist to store the data in the ELN, but it also allows to automatically record a lot of metadata—for example, the importation workflow can fill in information about the instrument (such as manufacturer, serial number, humidity, the temperature of the room) that is not always recorded in the output files of the measurements (see Figure 5 and Figure 6 for examples).

DATA PROCESSING After the data has been produced and imported into the ELN, most data need further analysis. Currently, chemists must switch between different, often proprietary, software to analyze their data. They might rely on the software provided by the instrument manufacturer to perform peak picking or baseline correction and then use another plotting tool to overlay the data. In an open-science vision, one would like to ensure that one not only has access to the data but, equally important, one can also reproduce the subsequent analyses. Likewise, if the chemistry community would embrace the view that the ELN has converted the data to a commonly agreed standard data form, the analysis tools become independent of a particular instrument or even characterization technique.

If one designs the platform with a common interface, ensures a modular architecture, and ensures a re-usability of the key components, one has the first step towards an ecosystem in which libraries are developed for specific tools that accelerate the workflow of chemists (Thesis 4). The modular nature would allow experts in one technique, for example, NMR spectroscopy develop tools that can then be reused by other ELNs. An example of this is the NMRium project⁸⁸ which is a reusable web component that can, with three lines of code, be plugged into another ELN system. To make this work, it is important that the components can talk with each other via standardized protocols.

In an open science vision, the code for these components should be open. One of the concerns regarding open source software is the danger that a project might “die out” if one maintainer leaves the project, whereas successful commercial software might seem to have the promise of continuity. However, there are many successful examples (such as Linux and Python) in which open-source projects are maintained by the community yet leave many options for commercial initiatives (for instance, support contracts and maintenance of a custom installation). Similarly, at universities, common analytical infrastructure (such as the routine NMR service) is often supported using institutional funding—a similar model might also be appropriate for the digital infrastructure. Importantly, open source code has the advantage that the underlying assumptions and equations for any analysis are documented, and everyone can verify, replicate, or even improve the analysis. And in contrast to closed source (commercial) tools, which have been discontinued because of a change in business interests, the development can be reanimated at any time, as the code is openly accessible and reusable.

PUBLICATION OF REUSABLE AND MACHINE-ACTIONABLE DATA The work of a scientist is not completed when all materials are synthesized and characterized. An essential part of the scientific process is the dissemination of the results to make sure that others can build on top of one’s work. Typically, we are used to thinking of “others” as other scientists in the same field. However, science is increasingly multi-disciplinary; hence non-specialists might also need to understand the data. Additionally, the move towards open science is a logical consequence of the notion that if the taxpayer paid for the research, the ownership of the research data should be the public, which can empower citizen (data) science.^{89,90} We can get a glimpse of the power of the re-use of data with the discoveries of Don Swanson, an

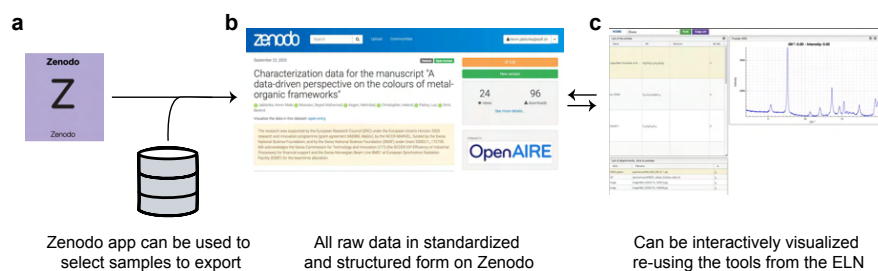


Figure 8: Example of the data flow from the ELN to the interactive visualization for the reader of a paper. Once all the chemicals are selected of which the synthesis and characterization data needs to be published, the ELN compiles the data and uploads the data to a repository (in this case Zenodo⁹¹). These data are not only machine-readable, but the data can be accessed through a browser, and a human reader can also use the same visualization tools as the authors of the article.³⁴ The implementation sketched in this figure is implemented in the open-source cheminfo ELN.

information scientist without formal training in medicine, who analyzed literature from the Medline database and found previously undiscovered knowledge like links between magnesium deficiency and migraine.⁹⁰ Nothing fundamental about chemistry prohibits us from leveraging such approaches to science.

Usually, however, in contrast to the publication of the article, the publication of all the scientific data on which this article is based is reduced to an afterthought. Most of us are still educated with the idea that we need to be selective about the data instead of embracing the idea that all scientific data we generate is an integral part of the science we do: Data is only published to fulfill the requirements of some journal policy or data management plan—without the reuse of the data in mind. This likely explains why many ELNs do not feature an option to export data to a repository.

In the open-science platform we propose, the publication of the scientific data is simply seen as one of the applications that can be applied to the data in the ELN. In such an application, the users can select the samples they want to publish and create an entry on a repository containing all the raw data. The application ensures that data are reported in a form that can be easily reused by other researchers and machines. For the chemists writing a publication, this means that they can provide a DOI to supplementary material and augment every figure with a link at which readers can interact with the raw data or download it for follow-up studies. Both the chemotion and cheminfo ELNs implement parts of this functionality. The cheminfo ELN exports data to the general-purpose Zenodo⁹¹ repository, whereas the chemotion ELN can export data to the chemotion repository,⁵³ which focuses on chemical synthesis and characterization data.

Similarly, an ELN might also allow importing entries from a repository. That means that researchers might import the entire lab notebook used to produce published results. Importantly, since the characterization data is also provided on the repository, the researchers would also have access to the original characterization data and might overlay them with their new results. To our knowledge, no ELN fully implements this automatic re-importation procedure.

1.3 DISCUSSION AND OUTLOOK

The open science platform we propose in this work provides a central hub for all the synthetic or analytical work of a chemist or materials scientist. Underpinning this platform are two common principles essential to making it truly open science,

such that it can benefit data-intensive research and address reproducibility problems (Thesis 1). First, FAIR data should be at the core of the platform; all data that enter the platform need to be converted to an open, structured, and standardized form with appropriate, linked metadata—this is the core functionality an ELN should provide (Thesis 2). Second, open science also ensures that other researchers can reproduce and build upon the results. Therefore, the platform should be able to export the data in a form that is machine-readable and -interpretable and can easily be re-used by other groups (Thesis 3). In addition, in an open science vision, one would also like to make the tools used to analyze the data available to anyone in the world interested in reproducing the results or reinterpreting the data. This leads to the notion that such a platform is ideally developed as modular open source infrastructure where the community can scrutinize, reuse, and improve the analysis code (Thesis 4).

The possibilities are unlimited if such a platform becomes widely used and supported by the community. The way we assess scientific work and credit scientific outputs has the potential to change. trusted timestamps can provide unique proofs of discovery, going beyond the compressed and delayed priority claim preprints can provide,⁹² and peers can continuously provide feedback about the raw data, the analysis, and the conclusions. An interesting form of making the full research record public, and hence open for feedback, has already been proposed in the context of open notebook science.⁹³ If this information is shared with the community, one can build a community-driven version of the *Organic Syntheses* journal where the verification of the results is done continuously by the community and not (only) in a lab of one of the members of the editorial board. Importantly, this version would also contain information about the attempts that did not work and, in this way, document the process, and the learnings, that led to the final result. If data is available in digital form, the peer review process can be supported with automated checks, for example, to verify the consistency of NMR assignments and highlight potential issues for peer reviewers.

The most important reason for embracing the approach described in this perspective is that it can change the way we do chemistry. Before the digital era, many of us were educated with the idea that if we publish all the data we generate, any normal human being will get lost in the sheer volume of data. Data-intensive science, however, fundamentally changed this point of view. With machine learning, we have the tools to analyze orders of magnitude more data than any normal human can process, discover correlations in millions of data points, and build predictive models.⁹⁴ For example, if we aim to synthesize some compound, a simple query in the collective ELN database might show for one synthesis route 100 “failed” reactions and two successful ones, while another route shows 90 successful and ten “failed” attempts—which is a clear indication which synthesis route one should try first. Undoubtedly, a very experienced chemist might have very good intuitions about what works and what does not. However, for a new student in the field, this collective knowledge now becomes accessible. Clearly, we can go beyond this simple search and try to harvest the collective knowledge generated by all chemists, using machine learning techniques to capture subtle correlations across the chemical space of millions of reactions carried out worldwide. In this respect, machine learning is not different from the experienced chemist; most likely, it can learn even more from “failed” and partially successful experiments as from the successful ones. But in contrast to the chemist, it typically needs large amounts of structured data—which we could easily generate in chemistry.

Another issue that the chemistry community faces with open data is that everyone agrees that there are benefits in making data reusable and in reporting “failed” experiments, but often there is hesitation from individual researchers to adopt this

behavior until all members of the community do so. The social sciences give us various possible solutions to this problem setting.^{90,95} One approach is some kind of compulsion. For example, the fact that the submission of DNA sequences is a condition for publication in the leading scientific journals of the field is seen as one of the reasons for the success of the GenBank database.⁹⁶ This, in turn, opened many doors for bioinformatics research. We also witnessed that in small groups that include leaders of the field, agreements like the “Bermuda Principles”, which require that DNA sequence data are automatically released in publicly accessible databases directly after the measurement, can be achieved. In chemistry, we have observed similar dynamics in crystallography, where Crystallographic Information Files (CIFs) must be deposited with the Cambridge Structural Database, where they are made freely accessible (and searchable) on publication. This led the European Commission to conclude that “the requirement from academic journals that authors provide data in support of their papers has proven to be potentially culture-changing, as has been the case in crystallography”.⁸⁶ What we can also learn from crystallography is that once some standards are adopted, automatic checks (such as checkCIF) can be implemented.

From the Structural Genomics Consortium (SGC) and related initiatives (for example, Open Source Malaria,⁶⁴ COVID Moonshot⁹⁷) we can learn that openness can also be enforced on the level of a consortium, for example, by requesting that members openly publish the protein structures and not to file patents for the research outputs. This public-private partnership (PPP) model seems to be successful because the private sector, which provides the funding and “chemical probes” (potent inhibitors of protein function), can guide the research—i.e., prioritize structures that should be solved—without disclosing the companies R&D priorities since the consortium anonymizes the “wish lists”.⁹⁸ The utility of such a consortium can best be seen at the pre-competitive stage (i.e., the early stages of drug discovery), where it can share risks, enhance collective learning, and avoid duplication in new areas of (basic) science.⁹⁹ This is particularly interesting in the case of “chemical probes”, which are best produced by experienced industrial, medicinal chemists. However, the industry would enormously profit if academia could use those probes to validate drug targets.¹⁰⁰ For this reason, the SGC makes them available as “open access” reagents—under the conditions that the research outputs are made available in the public domain. A similar “physical open access” approach is also pursued by the Molecule Archive of the Compound platform at the Karlsruhe Institute of Technology (KIT), which acts as a mediator for compound exchange: Synthetic chemists can “archive” their compounds (which increases their visibility), which can then be requested for biological screenings.¹⁰¹

Beyond those measures, we must change incentive structures by creating better ways to credit researchers for curating data. ELNs could help in this regard by storing the “credit” chain when data is imported and automatically append the citation when data sets are prepared for publication.

Beyond that, adapting this data-centric approach to chemistry requires changes in the university curriculum to raise the awareness of these new developments, as well as the need for and the promises of data curation. Ideally, open-science solutions like the infrastructure described here should already be introduced in the undergraduate curriculum. Students can record the results of their labs in ELNs, harvest the data in machine-learning classes, predict the IR spectrum they just measured in computational chemistry classes,³⁷ and use open notebooks to comment on and improve each others’ work.

The question that might still be open at this point is how realistic the widespread adoption of such an open-data platform across the chemistry community is. We argue that we have all the basic tools and technology in place. For many of the key

design aspects, we used examples from our own work, which are openly available, can be tried out by the community, and can be reused in other implementations. There are also several initiatives (see Table 8) that work on some of the aspects we emphasized in this perspective. One example is the German NFDI4Chem consortium,^{102,103} which is embedded in the larger German initiative for the creation of National Research Data Management Infrastructures (which also includes NFDI4Cat¹⁰⁴ for catalysis research and NFDI4Ing for the engineering sciences), and aims to “FAIRify” the full data lifecycle in chemistry. But we, as a community, also have to realize that we are in a phase where there is an “insane” number of initiatives, proposed data schemas, and ELNs. As a community, we face the task of embracing and connecting the efforts. Only if we succeed in making these tools interoperable will we be able to leverage the full potential of data and the digital age. One promising way forward will be the formation of data communities,¹⁰⁵ where experimentalists and ELN developers work together to develop a domain-specific (for example, porous materials, batteries) open science infrastructure by combining, extending, and polishing existing building blocks.

From our perspective, there are a concrete few steps that need to be implemented to reach this goal

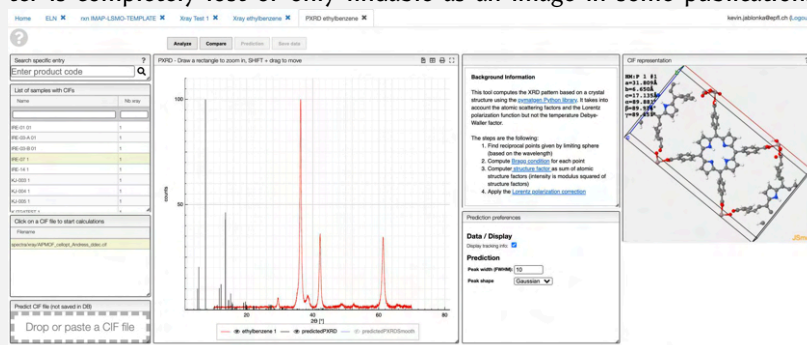
1. **The chemistry community should embrace its own existing standards and solutions.** We will only be able to make progress as a community if we start connecting and using existing solutions. The feedback can then be used to improve the tools. If we as a community never move beyond the stage of just proposing new formats or implementations—instead of using them in practice—we will not make any progress. This also requires that the existing tools are made reusable (i.e., packages are extracted from monolithic code bases and augmented with documentation) and shared on platforms such as GitHub.
2. **Journals need to make deposition of reusable raw data, where community standards exist, mandatory.** This is motivated by the success of the Bermuda agreement and the deposition of CIFs and is needed to address the collective action problem. Just using ELNs does not solve the problem. We also need to open our ELNs. Notably, this does not mean that data should be provided as PDFs but in a standard, machine-actionable form. Where community standards exist or are emerging, for example, as it is happening in the field of gas adsorption,¹⁰⁶ journals should start embracing those formats by requesting the deposition in a community repository.¹⁰⁵ The same holds for the basic characterization of organic compounds (NMR, IR, MS), where the chemotion repository already offers tools and curation that are reminiscent of the CSD. Importantly, disconnected data in different repositories can often only practically be used if they are linked. Therefore, for instance, the gas adsorption data on one community repository (such as the NIST/ARPA-E Database of Novel and Emerging Adsorbent Materials¹⁰⁷) needs to be linked, ideally using hyperlinks, to the crystal structure in the CSD.
3. **We need to embrace the publication of “failed” experiments.** Using a digital infrastructure, this can be easily done to tell the story of how the final result was reached. It also requires that we as a community realize that the outcome of an experiment is not a binary “is this a breakthrough or not” but simply an observation that is valuable and can be reported. For this to be successful, we must take care to properly acknowledge such datasets, for example, when we used them for data mining exercises, or they helped us to avoid some costly experiments.

4. **ELNs that do not allow exporting all data into an open machine-actionable form should be avoided.** This reflects the core of Thesis 2: The most important service an ELN can provide is removing the hassle of making data FAIR. This is not only to avoid losing access to the data if a license expires or being unable to build on previous work as it was in the “old-ELN” format, but it is also about being able to collaborate and share data with groups independent of the ELN. ELNs that just store data as provided, and might not even allow exporting this data, do not bring us closer to the goal of reusable data in chemistry.
5. **Data intensive research must enter our curricula.** “Open science” is gaining momentum in the chemistry community, and increasing numbers of researchers are engaging (to various extents). We need to raise awareness of these new developments already at the undergraduate level, use ELNs for our lab courses, and teach that open science is just science done right.^{108,109} For example, at EPFL, we teach machine learning and the use of ELNs in the same course and plan to couple the lab courses with data analysis exercises in the ELN. This also implies that our institutions need to provide faculty with appropriate support, for instance, via the campus library.¹¹⁰

To conclude, we would like to emphasize that the technology is here not only to facilitate the process of publishing data in a FAIR format to satisfy the sponsors but to ensure that the combination of chemical data, FAIR principles and openness gives scientists the possibility to harvest all data so that all chemists can have access to the collective knowledge of everybody’s successful, partly successful, and even “failed” experiments.

Box 3: Example of the online chemical processing of data

A common operation in materials science and inorganic chemistry is characterizing a material with powder x-ray diffraction. One then typically compares the measured spectrum with some reference, which might be a predicted pattern, a single crystal structure, an entry from a reference database, or a pattern from the past, for example, with a pattern measured by a student that left the group. In the worst case, the latter is completely lost or only findable as an image in some publications.



In the cheminfo ELN, the same interface can be used to compute an x-ray diffraction pattern based on any crystal structure in the database, overlay it with experimental patterns measured in the past in the research group, or deposited in the Computation-Ready, Experimental (CoRE)-MOF¹¹¹ or the crystallographic open database^{112,113} (see screenshot). A typical question in this context is whether a structure is a distorted analogue of a known structure.

When our experimental partners approached us with this question, we could

extend the toolbox in the ELN to allow the calculation of XRD patterns for distorted cells of reference crystal structures—we see this collaboration with experimentalists as a key to the success of an ELN platform. Similarly, one can link computational infrastructure to give experimentalists easy access to “routine” simulations.³⁵

Again, the tools are reusable by other researchers—in the form of the source code and a web service that exposes a REST-API that can be queried from other systems, such as other ELNs.

We envision that web services such as this can be an essential part of a platform where the chemical processing of data happens online. Indeed, different web services can be developed and maintained by research groups in their field of expertise (and in an appropriate programming language) and reused by the chemistry community on any platform with any programming language.

2

AN ECOSYSTEM FOR DIGITAL
RETICULAR CHEMISTRY

ABSTRACT The space of all plausible materials for a given application is so large that it cannot be explored using a brute-force approach. This is particularly the case for reticular chemistry, which provides materials designers with a practically infinite playground on different length scales. One promising approach to guide and accelerate the design of materials is machine learning. While there have been plenty of examples of the use of machine learning for reticular materials, progress in the field has stagnated. From our perspective, an important reason is that digital chemistry is still more an art than a science in which many parts are only accessible to experienced groups. To address this, we present mofdscribe: a software ecosystem that accompanies—seasoned as well as novice—digital chemists on all steps from ideation to model publication. While we optimized the tools for the challenges of reticular chemistry, most, if not all, can also be used for studies on non-reticular materials. This ecosystem allows for a more robust, comparable, and productive area of digital chemistry.

CITATION This chapter is a preprint version of our *In Focus* article: Jablonka, K. M. et al. *ACS Cent. Sci.* **2023**, 10.1021/acscentsci.2c01177.

CONTRIBUTION K.M.J implemented the software library, conducted the experiments, and wrote the manuscript with B.S.

2.1 INTRODUCTION

Reticular chemistry, the science of constructing extended crystalline structures from molecular building blocks, gives scientists a unique playground for material design and discovery as it gives access to a practically infinite-dimensional design space across many length scales: One can architect the pore, functionalize the building blocks, or even encode chemical sequences across unit cells.¹¹⁵ These possibilities made reticular chemistry one of the most active fields of modern chemistry with more than 100 000 structures collected in experimental databases.¹¹⁶ Exploring this entire design space by mere trial-and-error using brute-force computational screenings and iterative experimental testing is impossible. Similar to many other scientific domains,^{117–119} this realization gave rise to the notion of *digital reticular chemistry*,¹⁶ and in particular to the use of data-intensive research to aid the discovery and design of new reticular materials for any given application by learning predictive models from data.^{94,120} Thus far, machine-learning approaches have—among others—been used to predict gas adsorption properties,¹²¹ colors,³⁴ oxidation states,¹²² electronic properties,^{123,124} heat-capacities,¹²⁵ or synthesis conditions¹²⁶ as well as (water) stability of MOFs.^{127–129}

This work is motivated by the observation that the full potential of data science in this field has not yet been achieved. We argue that some critical bottlenecks limit our progress. Even though all works operate on the same class of materials and often use related machine-learning approaches, these works are hardly comparable or replicable and only implementable by experienced groups. This impediment is present across all stages of the machine-learning workflow. Researchers use different datasets to train and test their models—as we show, sometimes with significant data leakage¹³⁰—preventing direct comparison of modeling approaches. Further down the modeling pipeline, practitioners often use different implementations of the same technique to convert structures into feature vectors—or do not attempt to try different strategies due to implementation challenges. At the end of the modeling process, models need to be validated. However, also there, researchers use different protocols, and—as we discuss—not always the most meaningful ones. Together with the lack of platforms that compile the results obtained with different approaches, these bottlenecks make machine learning for reticular chemistry still more an art than a science.

For many machine learning applications, it has been observed that these problems can be overcome by providing a proper scientific ecosystem for the field: providing the basic building blocks for all the relevant steps in an easily accessible form.¹³¹ If such a software ecosystem is in place, users can radically accelerate the pace of innovation (as they can use interoperable building blocks and reuse others' work) while ensuring that their work contributes to the advancement of the field. In this work, we report a software ecosystem that aims to achieve this goal.

Our ecosystem provides machine learning-ready datasets, along with more than 40 reported and unreported featurization approaches, under a consistent API that enables rapid experimentation and makes those tools accessible to non-experts. Moreover, to facilitate consistent and meaningful evaluations of machine learning approaches, we also provide data splitters, as well as benchmarking tools that allow submission to a public leaderboard that is automatically updated upon submission.

Using materials design case studies, we illustrate the importance of these best practices, negligence of which can, in some cases, lead to the selection of models with much worse generalization performance.

Importantly, while our tools are optimized to address the challenges and opportunities of reticular chemistry, most, if not all, can be applied to other material classes. This also applies to our case studies, such as the one about the impact of data leak-

age.

2.2 RESULTS AND DISCUSSION

Machine learning studies typically need to go through multiple, often iterative stages, all supported by our mofdscribe software library.

1. *Collecting a dataset.* For machine learning efforts to be comparable, consistent datasets, along with measures that mitigate data leakage, are needed. In mofdscribe, we provide a consistent interface to multiple commonly used datasets^{111,116,132–134} as well as a completely new dataset of adsorption properties, complementing the QMOF database.^{123,124} Additionally, we implement measures to mitigate the effects of data leakage.
2. *Featurizing a material.* Most machine learning models only accept inputs of fixed shape. Therefore, structures (which generally have varying numbers of atoms) need to be converted into fixed-sized arrays. However, since some of these strategies have no reusable open-source implementations or existing ones are hard to combine, researchers seldom explore different featurization approaches. To address this, we implemented more than 40 different such featurization strategies that have been used in the literature as well as completely new ones.
3. *Splitting the datasets.* To estimate the generalization performance of a model, it needs to be evaluated on data it has not seen before (i.e., is independent of the training data) and, ideally, mimics the distribution of data the model will be used on.^{130,135} For this, one typically splits the dataset collected in step ① into multiple parts. However, as we show, the chosen strategy can have an important impact on model selection and interpretation of the results. Therefore, mofdscribe implements multiple reported and novel splitting strategies to ensure stringent model evaluation.
4. *Evaluating performance.* Moreover, to compare and evaluate models, we need to compute metrics.¹³⁶ However, as we argue below, practitioners tend to report commonly used metrics instead of ones that are actually relevant to the application. We showcase such a more relevant metric and implement it along with others in the mofdscribe package.
5. *Comparing the performance with the state-of-the-art.* For science to make progress, it is important to be able to compare with and build on top of others' results. In the current state of digital reticular chemistry, this is not possible. To address this, mofdscribe implements benchmarking tools that allow direct submission to task-specific leaderboards. Furthermore, the design of our benchmarking tool requires users also to share their hyperparameter optimization strategy.

2.2.1 Structure datasets

Many machine learning practitioners recognize benchmark sets as drivers of progress. For instance, researchers in image classification can easily compare the performance of competing approaches, as they can compare model performance on the same tasks on the same dataset (e.g., ImageNet¹³⁷).¹³⁸ Over the last years, similar benchmark

datasets have been reported for generative models for molecules¹³⁹ or quantum machine learning.^{140–144} However, there is currently no widely used reference set for machine learning on metal-organic frameworks (even though the QMOF dataset^{123,124} makes important steps towards this goal). As a first step towards more comparable machine learning for MOFs, our package implements a consistent interface for collections of structures (`StructureDatasets`), along with some corresponding properties (e.g., gas adsorption or electronic properties) with which all datasets can be used via the same interface. Our package implements reference datasets based on the QMOF database,^{123,124} the ARC database,¹³³ the BW database,^{27,132,145} the ARABG database,¹⁴⁶ as well as on a subset of the CoRE-MOF database.^{111,134}

A challenge with the currently existing datasets is that different properties are computed for different structures. However, for many learning applications, having multiple properties for the same structure can be useful. To address this, we used reproducible computational workflows¹⁴⁷ to compute diverse gas separation properties (CO_2 , CH_4 , H_2 , N_2 , O_2 isotherms; H_2S , H_2O , Kr, Xe Henry coefficient as well as parasitic energy for carbon capture from natural gas and a coal-fired power plant^{148,149}) for nearly seven thousand materials from the QMOF database (which contains many nonporous materials) and make them accessible via our `mofdscribe` package. To the best of our knowledge, this makes it the first database of some gas adsorption properties collected alongside many other properties (e.g., bandgap computed with different functionals) of the same structure. We intend to update the database in parallel with the QMOF database.

Data leakage

A pitfall for machine learning studies is data leakage, which means information from the test set is leaked into the training.¹³⁰ Often this can happen if, for instance, hyperparameters are tuned based on metrics computed on the holdout test set. However, data leakage can be much more subtle. For example, slight variations of the same structure might occur multiple times in one dataset. Machine learning based on data extracted from experimental crystallographic databases [such as the CSD¹⁵⁰ or the Crystallographic Open Database (COD)¹⁵¹] is particularly prone to this kind of data leakage as one structure can appear multiple times under different identifiers in the database. This can, for instance, be the case because there are different refinements for the same structure or because the measurement was performed at different temperatures. The presence of duplicates in MOF databases has been reported before^{152,153} but is seldom considered in machine learning studies. In Case Study 1, we illustrate that for MOFs data leakage is indeed a severe and perhaps underestimated problem.

To address this problem, `mofdscribe` implements computationally efficient heuristics that help with the deduplication of datasets. Those heuristics are based on the computation of hash strings of the periodic structure graph (so-called quotient graph),¹⁵⁴ which describes the connectivity of the atoms in the crystal (vertices being the atom positions and edges being the bonds). Since the structure graph does not directly depend on the exact atomic positions, structures with slightly different atomic positions (e.g., conformers) will share the same structure graph. While a check for graph-isomorphism would be the formally exact way to check for duplicated structure graphs, this can be computationally intensive, or even prohibitive, for large structure graphs as they are common for MOFs. Therefore, we use Weisfeiller-Lehman¹⁵⁵ hashes of different versions of the structure graphs (Figure 9), corresponding to increasingly tight definitions of duplication. We want to emphasize that while this deduplication strategy is a good default for most applications, it might be too strict for others as, for instance, open and closed forms of a framework will be counted as

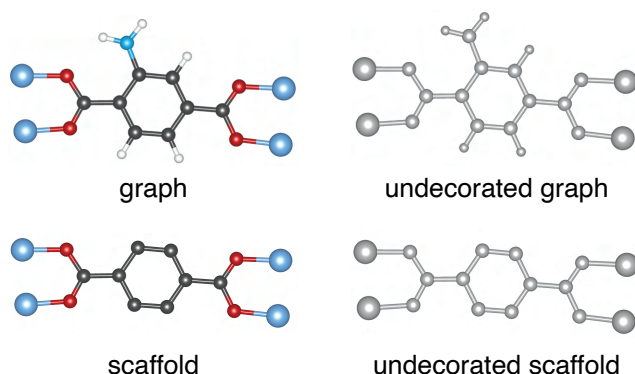
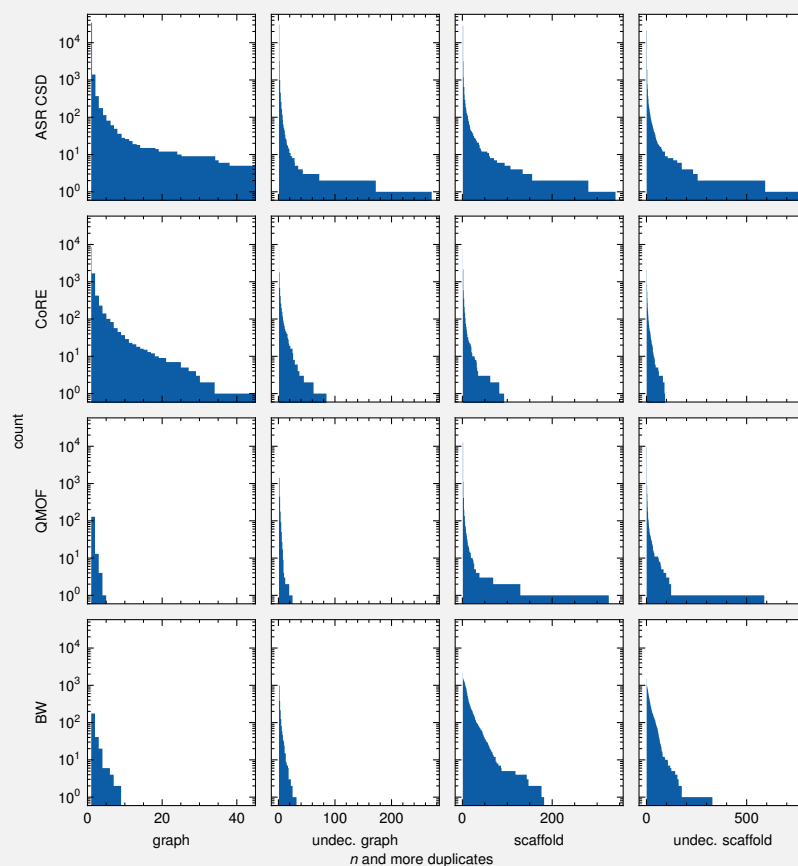


Figure 9: Levels of structure graph abstraction. **a** The (decorated) structure graph considers all atoms, bonds, and atomic numbers as “coloring” of the graph. Therefore, structures with slightly different geometries (e.g., experimental vs. DFT-optimized) but the same connectivity will be considered equivalent. (We recently used this definition to find duplicated structures in the CSD when matching structures with their isotherms.¹⁵⁶) **b** To find structures with the same connectivity but different coloring (e.g., Mg-MOF-74 and Ni-MOF-74), we can use the undecorated graph. **c** A harsher measure of structural similarity can be obtained by only considering the scaffold. Here we form the scaffold by breaking all so-called bridges. Bridges are edges (i.e., bonds) whose breakage leads to an increase in the number of connected components. Those are usually coordinated solvents, hydrogen atoms, functional groups, or other terminal atoms. Note that this definition of scaffold differs from the one used for (Bemis-Murcko) scaffold analysis of molecules.¹⁵⁷ Therefore, fluorine, chlorine, or amine-functionalized structures would all be treated equally. **d** Also here, we can remove the coloring to make, for instance, Ni-MOF-74-NH₂ equivalent to Mg-MOF-74-NH₂. To simplify the identification of duplicates, we use the Weisfeiler-Lehmann test to convert graphs into a hash string. While this test does not guarantee isomorphism, the resulting computational advances drastically outweigh the lack of theoretical guarantees.

duplicates. Hence, this automatic deduplication can be disabled and customized in mofdscribe.

Case Study 1: The impact of data leakage One can argue that there will be a few errors in any large data set. However, the presence of duplicates can cause serious issues for model evaluation.¹³⁰ In this case study, we show that this is a severe problem for reticular chemistry.

Let us start by investigating the number of duplicates in commonly used databases of experimental and hypothetical MOF structures. In the case of the experimental databases, it is important to realize that if the same structure has been refined multiple times or measured under different temperatures or with different unbound solvents, it will appear under multiple CSD reference codes. From a machine learning point of view, however, these materials are too similar to appear in both train and test sets (as for many applications such as high-pressure gas storage, the model could then make a perfect prediction by just remembering the appropriate training data).

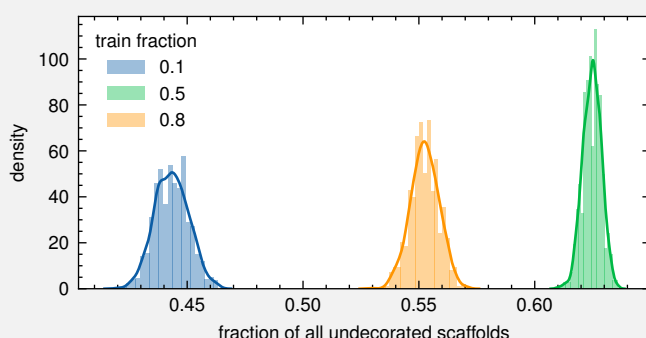


Box Fig. 1 | Duplicates across MOF databases. We show the number of duplicates as inverse cumulative histograms, i.e., the bar heights indicate how often we find n or more times a duplicated graph type. In the columns, we show increasingly general (i.e., more structures are considered as duplicates) definitions of duplicates. The rows show the counts for different databases: The all-solvent-removed (ASR-CSD) subset of the 2019 MOF subset of the CSD, the CoRE database, the QMOF database, and the Boyd-Woo database (BW) of hypothetical MOFs.

Using increasingly general (i.e., more structures are considered as duplicates) definitions of duplicates (Figure 9), we analyze how many matching structures we find in the all-solvents removed (ASR) subset of the CSD MOF subset (2019), and the CoRE and QMOF databases. The inverse cumulative histograms below plot how often we find n or more identical hashes (e.g., at $n = 10$, the count represents the number of structures with 10 or more identical hashes). The “graph” strategy considers all structures which share the same connectivity and atom types as identical. The “undecorated graph” (undec. graph) strategy does not consider the atom types. The “scaffold” strategy removes all functional groups, solvents, hydrogens, and terminal atoms from consideration (formally, all subgraphs connected via bridges). Again, we can also remove the atom types from consideration. As expected, we see an increase in the larger number of duplicate counts from left to right. This analysis shows that, for instance, the Co-CPO-74 structure appears 114 times in the ASR CSD MOF subset. This implies that this structure will likely appear both in the training and test set. Importantly, this is not the only structure with many duplicates; there are of the order of 100 struc-

tures in which structure graph appears more than ten times in the CoRE MOF database—and hence do not contribute to a meaningful measure of the generalization performance of the model.

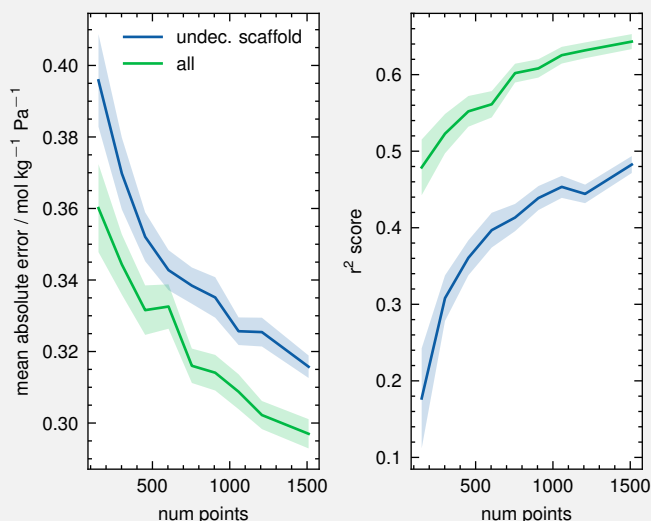
Analyzing a database of in-silico assembled MOF structures (BW),²⁷ we find notably many scaffold duplicates: Among nearly 20 thousand structures, we only find 1584 unique undecorated scaffolds. Many in-silico MOF assembly approaches enumerate all possible combinations of building blocks, nets, and functional groups. While this approach can give a lot of detailed insights into regions of chemical space, it will also give rise to many very similar structures that can lead to the violation of the assumption of independence between training and test set. To illustrate this, we can simulate a thousand random train/test splits (as commonly done) and measure how often a scaffold occurs both in the training and test set. For practical train/test ratios, the majority of scaffolds (e.g., 55 % for a train/test ratio of 0.8/0.2) will be found in both training and test set.



Box Fig. 2 | Likelihood of having the same scaffold in train and test set.

Using the BW dataset implemented in mofdscribe we perform 1000 random train/test split for different train/test ratios and count how often we find an undecorated scaffold hash in both the training and test set. The figure indicates that for commonly used train/test ratios, most scaffolds will be found in both training and test set.

To showcase the potential impact of such data leakage on model evaluation, we computed learning curves (for CO₂ Henry coefficients in the BW database as implemented in mofdscribe) for two deduplication levels: No deduplication and removal of identical undecorated scaffolds. If there was no data leakage, the learning curves should be similar. However, for the deduplicated datasets, we observe that the initial learning is faster (presumably because of a higher information density in deduplicated datasets) but much lower than for the dataset containing undecorated scaffold duplicates. The increase in performance we observe for the data set with duplicates is most likely caused by data leakage; the test set contains many structures that are only marginally different from the training set, and hence if we remove these duplicates from our training and test, our error *worsens* significantly. Depending on the strictness of the duplicate definition (Figure 9), one might see—with the same train and test set sizes—drastically larger errors. Moreover, it is essential to realize that in our case study, removing duplicates increased the learning (steeper learning curves; in fact, this can sometimes lead to better models) and led to a more faithful measure of generalization performance.



Box Fig. 3 | Learning curves with and without duplicated identical scaffolds.

For this experiment, we trained gradient-boosted decision trees using the default feature set (currently including histograms of persistence diagrams, AMD, geometric properties, APRDF) implemented in mofdscribe on the BW database subset used in Moosavi et al.¹³² to predict the CO₂ Henry coefficient (which we reuse from Moosavi et al.¹³²). We used a train/test/valid split of 0.7/0.2/0.1 and performed the experiment 100 times. The shaded areas indicate 95 % confidence intervals around the mean.

2.2.2 Featurizing reticular materials

Before using the datasets to train a model, one typically needs to convert the structures into fixed-length feature vectors. This is required because most machine learning algorithms can only operate on fixed-sized inputs. For instance, we can envision that we want to predict the gravimetric gas uptake of a MOF—a property that should be the same regardless of whether we create a supercell, translate, or rotate the unit cell. That is, we need a function with which the cells of different sizes, or ordering of atoms, are mapped to the same feature vector. This example already illustrates that such a conversion of a structure into a feature vector is not unique and is always connected to certain, often hidden, assumptions. Ideally, those assumptions reflect a physical or chemical understanding of the system and act as inductive biases that help the learning algorithm.¹⁵⁸ But any approximation always limits the expressivity of the model, and many featurization approaches neglect—by design—certain aspects of a given system. Additionally, there are always certain design choices (such as the numbers that are used to encode chemical elements) that are not ideal for all applications. In mofdscribe we propagate those approximations (such as elemental encodings or aggregations) to the user and, therefore, allow tuning those parameters to increase predictive performance.

A key design aspect for featurizers is the length scale they operate on (Figure 10). In mofdscribe, we distinguish featurizers operating on the local, atom-centered neighborhoods, the building units (BU), and the full, global structure. Depending on the learning tasks, different scales will be more relevant. For instance, for gas separations, we need to describe the pore’s textural properties (global) and the building unit’s chemistry (BU/atom-centered). Therefore, it is important that featurizers op-

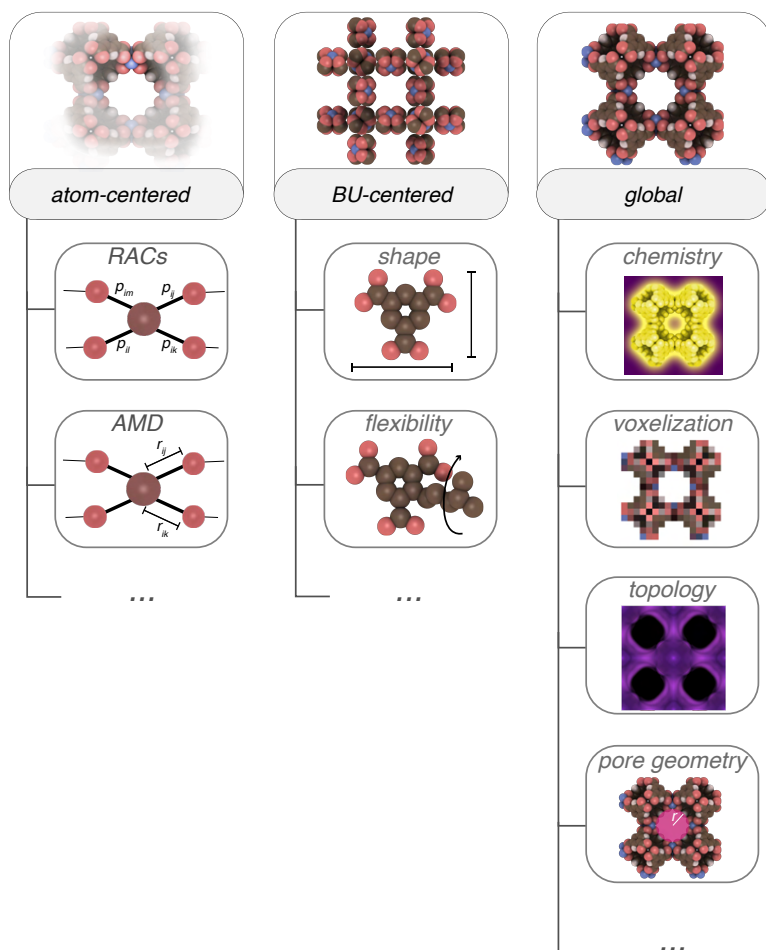


Figure 10: Overview of featurizer types implemented in mofdscribe. We distinguish three scopes on which featurizers operate: atom-centered, building unit (BU)-centered, and global features. Note that mofdscribe is interoperable with matminer, wherefore featurizers implemented in matminer can be used with those implemented in mofdscribe.¹⁵⁹ For a full overview of implemented featurizers, see Table 2

```

from matminer.featurizers.structure import JarvisCFID
from pymatgen.core import Structure
from rdkit.Chem.Descriptors3D import Asphericity

from mofdscribe.featurizers.topology import PHImage
from mofdscribe.featurizers.pore import PoreSize
from mofdscribe.featurizers.bu import RDKitAdaptor, BUFeaturizer
from mofdscribe.featurizers.chemistry import RACs
from mofdscribe.featurizers.base import MOFMultipleFeaturizer

s = Structure.from_file(<path to cif>)

featurizer = MOFMultipleFeaturizer[
    RACs(),
    PHImage(),
    PoreSize(),
    SurfaceArea(probe_radius = "C6H6"),
    SurfaceArea(probe_radius = 1.2),
    JarvisCFID(),
    BUFeaturizer(RDKitAdaptor(Asphericity, ["asphericity"]))
]

features = featurizer.featurize(s)
labels = featurizer.feature_labels()
citations = featurizer.citations()

```

Listing 1: Complete featurization usage example. The featurizers in mofdscribe can be easily combined with the ones implemented in matminer. All featurizers also share the same utility methods, such as `citation` and `feature_labels`, and can be computed for multiple structures using `featurize_many`. The `MOFMultipleFeaturizer` additionally provides the option to compute the primitive cell for each structure before featurization.

erating on different scales can easily be combined. In mofdscribe, all featurizers can be used in the same way and combined as needed, thereby enabling rapid experimentation. To make this possible, mofdscribe uses the featurizer design pattern popularized by the matminer package (see Listing 1 in which we compute a feature vector for a MOF by combining featurizers from all scopes). This design pattern, which bears similarities to the `sklearn` API,¹⁶⁰ ensures consistency across how different featurizers are used and, in this way, enables composability and also makes them accessible to non-experts. By building on top of the matminer building blocks, mofdscribe is also fully interoperable with the featurizers implemented in the matminer library. For instance, featurizers such as the matminer’s `SiteStatsFingerprint` can be seamlessly used to separately featurize framework and guest molecules using the `HostGuestFeaturizer` implemented in mofdscribe.

LOCALITY APPROXIMATION The most commonly used assumption in machine learning for chemistry and materials is the locality approximation. In practice, this assumes that a property does not depend on the entire crystal but that the main contributions are from the local environment (which can be justified based on the principle of “nearsightedness of electronic matter”¹⁶¹). For example, in our model for the oxidation state of the metal in a MOF,¹²² the features are computed for the metal and


```

racs_featurizer = RACS(
    prop_agg = ("avg", "product", "diff", "sum"),
    corr_agg = ("avg", "sum", "range"),
    bb_agg = ("avg", "sum", "range"),
)

```

Listing 2: Example of using aggregations in mofdscribe. Many featurizers compute more than one feature vector per structure; for instance, one feature vector per atom. In this case, the data must be further processed to construct one fixed-size feature vector per structure. To ensure that the resulting feature vectors are permutation invariant (that is, do not depend on the arbitrary numbering of atoms in the structure), one typically uses aggregation functions such as the average, sum, maximum, or minimum. At every point where aggregations are computed, we allow users to customize the ones used. Users can simply specify the desired ones as a tuple of strings; for example, ("median", "range", "geom_av") would aggregate the features using the median, range, and geometric average.

the atoms of the linkers surrounding the metal. By reducing the learning problem to local environments, the locality approximation allows a model trained on small fragments to generalize to large structures (which are harder to sample as there are combinatorially more of them).¹⁶² For reticular materials, this approximation is widely used as part of the feature set (which is often supplemented with global features, see below) via revised autocorrelation functions (RACs),^{132,163} smooth overlap of atomic positions (SOAP) fingerprints,¹⁶⁴ local geometry descriptors,^{122,165} or the average mean distance descriptor by Widdowson et al.¹⁶⁶ For all those atom-centered descriptors, one can compute N descriptors for a structure with N atoms. Since different materials will have different numbers of atoms N in their unit cells, one typically needs to perform an aggregation operation, such as computing the arithmetic mean of all the atom-centered feature vectors, to construct a fixed-length descriptor that is permutation invariant. The latter is important since we do not want our descriptors to change when we change the (arbitrary) numbering of atoms (that is only an artifact of digitally encoding materials). Of course, there is not one ideal choice for the aggregation operation. One might benefit from additional expressivity by using multiple aggregations, for instance, the standard deviation alongside the arithmetic mean or other Pythagorean means or robust measures (e.g., trimean, mean absolute deviation). Therefore, in our library, the user can—where applicable—simply provide all aggregation combinations of interest (see Listing 2 for an example), and mofdscribe will compute them all. As shown in Figure 82, this generalization of established descriptors (such as the AMD proposed by Widdowson et al.¹⁶⁶) can lead to large (>20 %) improvements in predictive performance on material property prediction tasks. By exposing all these options, mofdscribe makes these approximations visible to the users and allows users to tune them for better predictive performance.

Since models do not know the periodic table, the nature of the element types in a given structure needs to be encoded numerically. As in the case of RACs, this is often done using element properties such as atomic number or electronegativity. However, it is well known that some encodings, such as atomic numbers, lose the clustering of elements according to their periodic properties, which can be an important inductive bias for a machine-learning model. Therefore, mofdscribe allows users to flexibly choose from a wide variety of elemental properties and other encodings such as the (modified) Pettifor scales that have been shown to better capture similarities of elements across the periodic table.^{167–169} For instance, Pettifor scales can be thought of as the “optimal one-dimensional periodic table”.¹⁶⁹ similar ele-

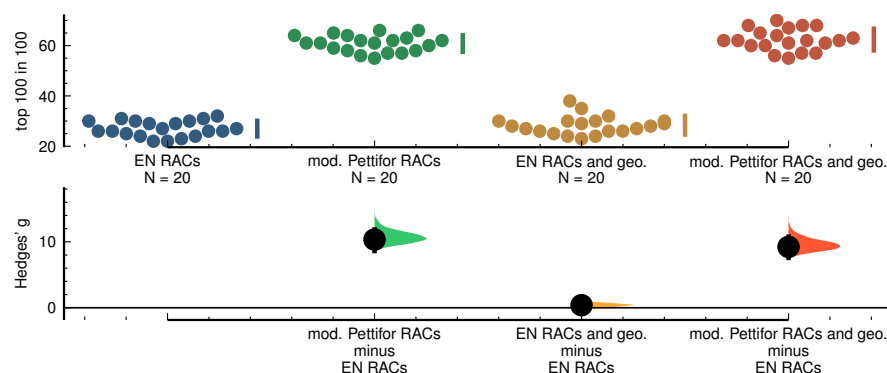


Figure 11: Impact of element encodings on predictive performance. Here, we compare revised autocorrelation functions with different element encodings as input for a gradient-boosted decision tree. We train the model on CoRE-MOF data reported by Moosavi et al.¹³² to predict the logarithm of the CO₂ Henry coefficient. The top row shows the number of top-100 materials we retrieve in the top-100 predictions of 20 independent runs. The bottom row shows estimated Hedges' g effect sizes (a suitable effect size metric in the case of little data¹⁷⁰ which can be thought of as a normalized mean difference) with respect to the performance of the RACs using electronegativity (EN) as element encoding. This figure also compares the performance of feature sets augmented with scalar geometric properties (pore diameters, accessible surface area, and void fraction). We find similar very large effects using other metrics.

ments are neighboring in this representation (in the original scale with the goal to achieve an optimal map of the stability of AB compounds). The impact the choice of encoding can have is shown in Figure 11, where we find improvements of over 50 % by using the modified Pettifor scale¹⁶⁹ for encoding elements in revised autocorrelation functions (in contrast to using the electronegativity). Also here, mofdscribe exposes these encoding options on all featurizers where they apply. This makes this approximation visible and allows users to tune the encodings to increase their models' predictive performance or interpretability.

BUILDING UNIT CENTERED DESCRIPTORS The defining feature of reticular chemistry is the tinker-toy principle, i.e., the construction of extensive crystal structures from small molecular building blocks. Interestingly, this principle is seldom exploited in machine learning studies for reticular materials such as MOFs. One possible explanation for this is that it is not trivial to extract the building blocks from a crystal structure into a form such that they can be used with featurizers that are conventionally used for molecules (e.g., as the ones implemented in the RDKit program¹⁷¹). To facilitate BU-centered featurization we implement an adaptor that can convert any featurization function that accepts a molecule object from the RDKit library (that can easily be generated from a SMILES string) into one that can be used along with all other mofdscribe featurizers. Importantly, to allow the decomposition of MOFs into their building blocks, we also release a library, called *moffragmentor*, that analyzes the structure graph to decompose MOF structures into their building blocks (i.e., metal cluster(s) and linker(s) and possible bound/unbound solvent, algorithm described in the Appendix). In contrast to existing tools such as *mofid*¹⁵² and *mBUD*,¹⁷² *moffragmentor* makes them accessible from an object-oriented interface. If a user provides a MOF structure into a featurizer for molecules (e.g., a conformer counter), mofdscribe by default fragments the MOF into building blocks and computes the features separately for each building block. The importance of this step is

that once a MOF is decomposed into building blocks, we can also generate descriptors that further characterize these building blocks. For example, descriptors related to the flexibility of the linker, such as the number of accessible conformers.¹⁷³ This is an example of a descriptor that could not be easily accessed otherwise but might help digital reticular chemists address questions (e.g., about crystallization) they could not easily address before. Importantly, all featurizers, e.g., also the SOAP fingerprint, can be used in this setting to compute more meaningful aggregations (in contrast to averages over the full structure). In this process, we do not enforce the use of our `moffragmentor` library, as users can bypass this step by providing their own building blocks. For example, users can provide `pymatgen`¹⁷⁴ `Molecule` objects that they obtain by deconstructing MOFs with other tools such as `mofid`¹⁵² or `mBUD`.¹⁷²

GLOBAL DESCRIPTORS In porous materials, many properties—such as the pore size and shape or overall composition—are not directly correlated to atomic environments or their building blocks. Therefore, local descriptors can only implicitly, via large cutoffs, or not at all, represent such properties.

For this reason, practitioners often use local feature sets (e.g., RACs) along with global ones; most commonly with scalars describing the pore geometry (e.g., pore volumes, surface areas, accessible volumes).^{175,176} However, additional vector-valued or count-based descriptors^{177,178} can be used to describe the pore geometry and might be more expressive than scalar descriptors but are seldom used in machine learning studies for reticular materials.

Describing the shape and chemistry: persistent homology As an alternative to the aforementioned pore geometry descriptors, the use of topological data analysis has been proposed to capture the shape of materials.^{179,180} Topological data analysis can be used to obtain features that are invariant to a continuous transformation of the material structure. Persistent homology, a branch of topological data analysis, captures all the topological information underlying a given point cloud, such as the geometric coordinates of a material. Given a set of points (i.e., atom positions), we can obtain a so-called filtration (e.g., Vietoris-Rips) by continuously increasing the radius of these points to get a family of nested unions of spheres. Persistent homology then tracks the appearance and disappearance of topological features (such as channels and voids) in this filtration. The radius at which a feature appears is the “birth,” while the radius at which a feature disappears is the “death.” The *persistence* of a topological feature is the difference between birth and death, which acts as a measure of how prominent a given topological feature is. The set of all birth-death points is called a persistence diagram. As reticular materials have many channels and voids, persistent homology provides a holistic approach to capturing these topological features. Persistence diagrams are a multiset of birth-death points in the extended plane, and each material can have a different number of points in its persistence diagram. Since most machine learning models operate on points in a fixed-dimensional Euclidean space, one needs to vectorize these diagrams into fixed-size arrays. One can accomplish this, for instance, by computing persistence images, which can be thought of as smoothed versions of persistence diagrams and have been used before in materials science.^{181,182} However, a challenge with this representation is that it is often very high-dimensional. Due to the curse of dimensionality, this can lead to learning problems (particularly in low-data regimes). Alternatively, we implemented a vectorization method that approximates persistence diagrams using Gaussian mixture models.^{183,184} This allows for low-dimensional representations that can still provide approximate Wasserstein distances (conventionally used to measure distances, i.e., a proxy for the difference between persistence diagrams). Additionally,

we also implement a simple vectorization as a 2D histogram.

Topological data analysis captures the geometry of chemical structures and materials, but these systems also have rich chemical information, as they are composed of different atoms. Thus, it is important to incorporate this chemical information into the representation—otherwise, materials with the same connectivity but different elements (for instance, Mg-MOF-74 vs. Ni-MOF-74) would be treated the same way, and a model would predict the same properties. To account for chemical information in a highly flexible way, which therefore can adjust to the amount of data available, mofdscribe allows decomposing the structures into structures that contain only certain elements (Figure 12). By default, for instance, mofdscribe will perform the persistent homology analysis on the full structure, the metal substructure, the organic substructure, and the halogen substructure. However, users can customize the substructures that mofdscribe considers and tailor the featurization to the task at hand. As Figure 13 shows, the inclusion of chemical information consistently increases predictive performance on MOF property prediction tasks (in our test cases by up to 20 %). We use the same approach to make the average minimum distance fingerprint proposed by Widdowson et al.¹⁶⁶ chemistry-aware (and observe similar improvements on benchmark tasks there). Additionally, we also allow users to encode chemistry using so-called *weighted* alpha shapes. In this case, in addition to the coordinates, the (atomic) radii of different elements are used for persistence diagram construction to distinguish between different atom types.

MOF tomography Some of the most successful applications of machine learning have been in computer vision.¹⁸⁷ The primary reason for this is that while images contain a wealth of information, it is hard to extract good features that can then be fed into a model (e.g., training a linear regressor on a flat vector of all the pixel values will not work). Convolutional neural networks (CNNs) and related architectures are *trainable* feature extractors.¹⁸⁸ That means, given some data, they can learn to extract the most predictive features (thereby enabling the use of techniques such as transfer learning). However, it is not obvious how one can convert structures, which might have a varying number of atoms and which unit cells might be skewed, into “rectangular” image tensors of fixed size. Additionally, one also needs to consider that one would also like to encode the chemistry of a given material. In mofdscribe, we implement featurizers that voxelize approximately cubic supercells of crystal structures into 3D images (which one could then process using a 3D CNN).^{189,190} Also for these featurizers, we allow the users to use aggregations of custom properties (e.g., Pettifor scale, electronegativities, atomic numbers) as the value for the voxels instead of just binary indication of occupied/unoccupied. Of course, for example, for low-data applications, our approach can only encode the geometry as a binary encoding, density, or using a truncated distance function. These features will, as initial results in the literature indicate,^{189–191} allow for using state-of-the-art computer vision models (including self-supervised pre-training and transfer learning) on reticular materials.

2.2.3 Consistent model evaluation and benchmarking

Having datasets and standard implementations of featurization algorithms is not all that is needed to make machine learning for reticular materials routine and comparable.^{192–194} To reach standard practices¹⁹⁵ for digital reticular chemistry, we must also address model validation and comparison.

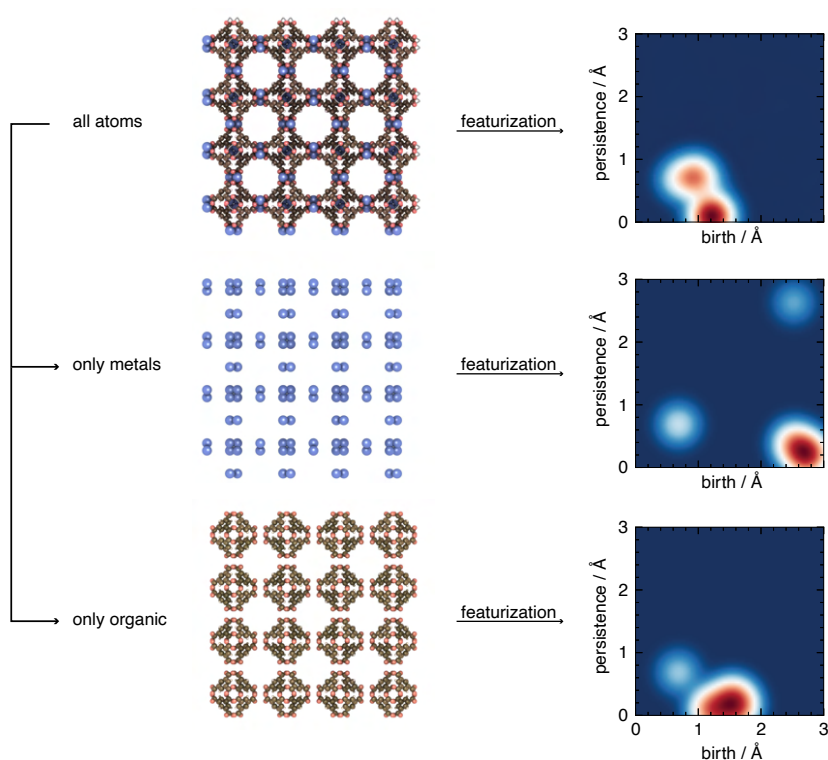


Figure 12: Adding chemical resolution to geometric descriptors. Descriptors, such as the ones derived from topological data analysis, operate on the full structure (top row) and capture the geometry and connectivity of a material described by its atomic coordinates. In *mofdscribe*, we allow users also to incorporate information from different atom types (i.e., not treat all atom types the same way). Users can customize the extent to which they want to lift this many-to-one mapping by adding channels for different atom types. By default, for instance, the descriptors from topological data analysis are computed for all atoms, the metallic substructure, and the organic substructure—all yielding different topological signatures, as evident from the persistent images.

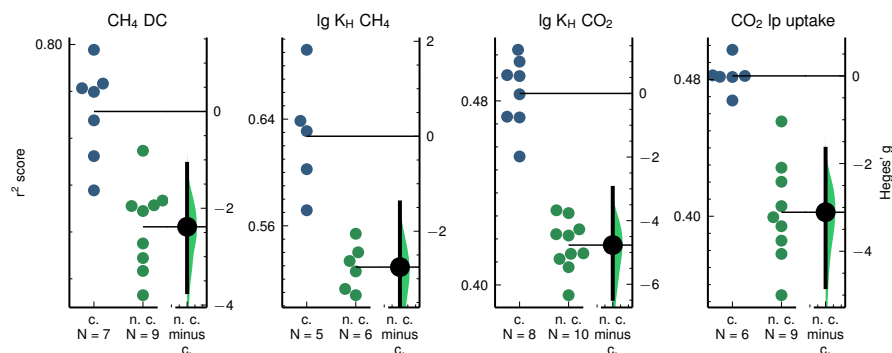


Figure 13: Persistent homology vectors with and without chemical information. In this plot, we use the vectorization of persistent diagrams using Gaussian mixture models with and without chemical information (here, we consider the C-H-N-O, halogen, and metal substructures). For this analysis, we optimize the *full* pipeline (including pre-processing and the model) using automated machine learning.¹⁸⁵ The plots visualize effect sizes in terms of Hedges' g . The points indicate the coefficient of determination (r^2 on a holdout test set, shown on the left axes) of the models trained to predict the methane deliverable capacity (DC), Henry coefficient (K_H), as well as the CO_2 Henry coefficient and low pressure (l.p.) uptake. The blue points are for the model trained with chemistry (c.) information; the green ones indicate the coefficients of determination of the models trained without chemistry information (n.c.). To quantify the effect, we bootstrap the Hedges' g (a suitable effect size metric in the case of little data,¹⁷⁰ shown on the right axes) and show it with a kernel density estimate. In all cases, the addition of chemistry shows very large effects.¹⁸⁶

Estimating material discovery ability

Many machine learning models are built to be useful for materials *discovery*. Discovery implies predicting something *unknown*. However, the common practice of using a random train/test split does not necessarily measure the model performance in predicting the *unknown*. First, a simple random train/test split cannot account for the fact that many databases contain very similar structures (see Case Study 1). As we have shown, (hypothetical) databases often contain multiple structures that are only different in the type or position of one functional group. Dividing such a group of structures with identical scaffolds across the train and test set will not give a faithful measure of the generalization ability of the model as one of the main assumptions of testing is violated—train and test set are not entirely independent of each other (one might see the functionalized graphs as children of the same scaffold). Second, for practical applications, there will almost always be a data shift; that is, the data distribution the model will be used with will be different from the distribution the model has been trained on. For example, it is well known that structures in hypothetical databases do not have the same distribution (e.g., lower density, less metal diversity) as structures in experimental databases.^{132,196} This has already been recognized by others such as Meredig et al.¹⁹⁷ who utilized leave-one-cluster-out cross-validation to estimate extrapolation performance (or Xiong et al.¹⁹⁸ using k -fold forward cross-validation). A similar approach, in which one clusters the principal components of the data into k clusters and trains on $k - 1$ clusters and tests on the cluster that was not used for training, is implemented in mofdscribe. As each cluster has specific properties, this method tests how well the model can extrapolate to new properties. If we repeat this procedure for every cluster, we can get an overall measure of the robustness and extrapolation ability.

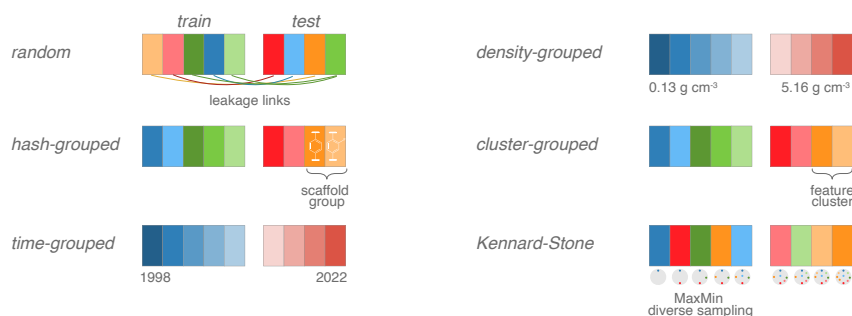


Figure 14: Illustration of some splitting approaches implemented in mofdscribe. For the validation of a machine learning model, it needs to be tested on data it has not seen before. To create sets of such unseen data, datasets are typically split into subsets for training, testing (and validation).⁹⁴ The conventional approach is to perform this split randomly. This, however, might not be a good approximation of real-world use with dependent/grouped data, distribution shift, or lead to problems in the case of imbalanced data. Therefore, mofdscribe implements various splitting strategies that operate on (structural) features or other metadata. For instance, we extracted the publication date for all structures we could trace back to the CSD and hence allow performing a time-based split. Alternatively, one can use structural features to ensure equal distribution of the features in different splits (stratification) or mimic a distribution shift/extrapolation case by forcing different groups into different folds of a cross-validation scheme. The easiest example is to use the density; however, one can also cluster (we use *k*-means clustering after principal component analysis (PCA)) on features, e.g., computed using mofdscribe). Moreover, we also implement a splitting strategy inspired by the scaffold splits sometimes used for molecules²⁰⁰ for reticular molecules via our hash strategies (Figure 9). This allows grouping structures with the same connectivity or backbone into the same fold. A different strategy is using Kennard-Stone sampling,²⁰¹ to ensure that the training set is maximally diverse.

Additionally, we recognize that one interesting benefit of working with experimental data is that we know when a given structure was first reported; the data is time-stamped.¹⁹⁹ Inspired by common practice in time-series forecasting, we hence can ask “could we predict the performance of materials discovered after year *X* if we only trained on materials discovered before year *X*?” In particular, we can measure how many of the top *k* materials we can recover in the top *n* predictions by the model. Using mofdscribe, this question can easily be answered.

Importantly, a time-based split is not the only feasible splitting strategy—and, depending on the use case, might not be the best option (or might not be applicable if no timestamps are available). Therefore, to further ensure that thorough model evaluation becomes routine for digital reticular chemistry, mofdscribe also implements, inspired by the DeepChem library,²⁰² a variety of `Splitter` classes (Figure 14, Listing 3 for a usage example). The `Splitter` classes either take a dataset that users can define based on their structures or a built-in dataset and can produce splits (holdout or *k*-fold cross-validation) following different strategies.

The impact such splitting strategies can have on model selection is shown in Case Study 2, where one can see that the average generalization performance of models significantly depends on the splitting strategy. Overall, we find that the optimal split depends on the task at hand—but typically is not the conventional random split. Given the redundancy in scaffolds in MOF databases, we urge practitioners to use grouped cross-validation.

As a utility to quantify the “difficulty” of the validation strategy, mofdscribe also implements a helper method that performs adversarial validation.^{203,204} Adversarial

```

from mofdscribe.splitters import HashSplitter, TimeSplitter
from mofdscribe.datasets import CoREDataset

time_splitter = TimeSplitter(CoREDataset())
scaffold_splitter = HashSplitter(CoREDataset())
train, valid, test = time_splitter.get_train_valid_test_splits(
    train_frac = 0.8, valid_frac = 0.1, test_frac = 0.1)

folds = time_splitter.k_fold_split(k = 5)

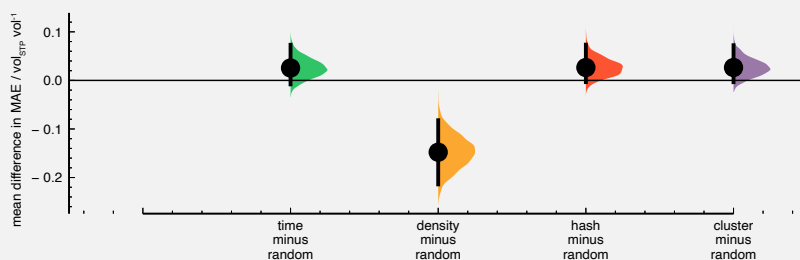
```

Listing 3: Example of the use of Splitters. The datasets implemented in mofdscribe already provide the relevant information for the splitters (e.g., times, hashes, densities). If the splitters are used on other structures, e.g., custom in-silico assembled MOFs, this information will be computed, if possible on the fly, or can be provided by the user. Note that the datasets, by default, are deduplicated based on the graph hash.

validation is a technique that has been popularized in data science competitions to measure—with only one number—the difference between training and test distribution but also to identify the most relevant features for a potential difference. For this, one simply trains a classifier to distinguish training from test examples. If the area under the receiver-operating curve (receiver-operating characteristic (ROC)-area under the ROC curve (AUC)) is close to 0.5, the classifier fails to distinguish the two datasets. However, if it does not (i.e., ROC-AUC close to 1), analyzing the feature importance can reveal the most relevant features contributing to the difference (which one might decide to remove to improve generalization). An application of this concept is shown in Case Study 2.

Case Study 2: The impact of splitting strategies To investigate the impact of splitting strategies, we train models on experimental data (for which we can also perform a time-based split) and then evaluate how well the models generalize to hypothetical materials.

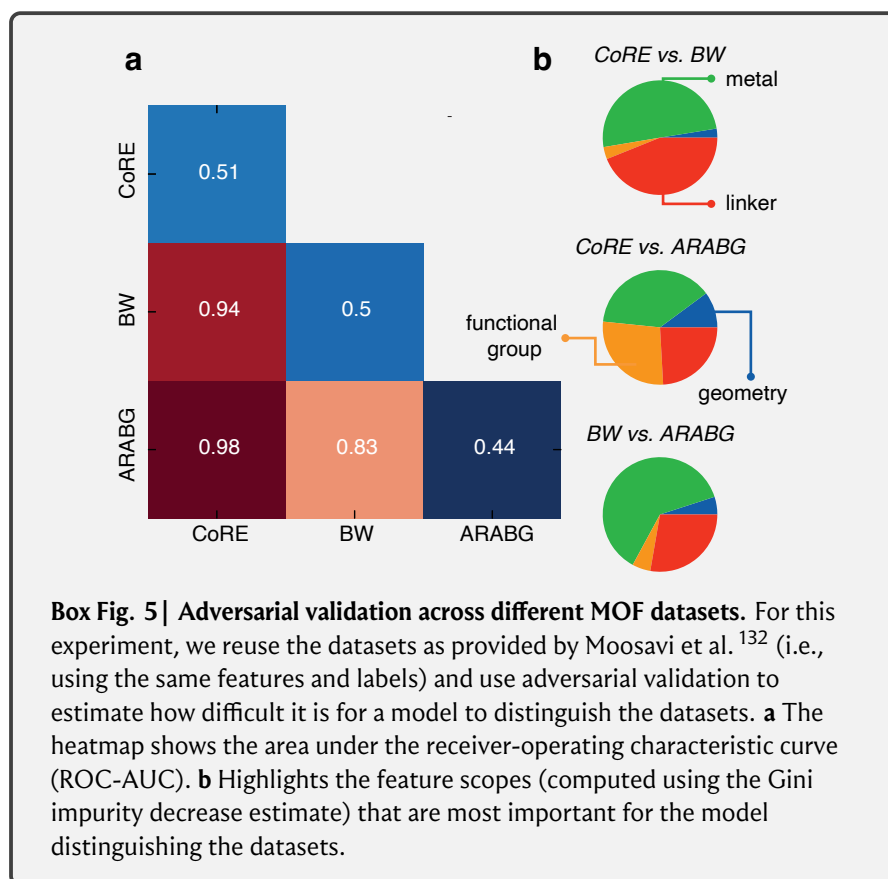
We trained the gradient-boosted decision tree models using the default feature set of CoRE dataset in mofdscribe (currently including histograms of persistence diagrams, average minimum distance (AMD), geometric properties, atomic-property labeled radial distribution function (APRDF)) to predict the methane deliverable capacity. For all experiments, we remove duplicates, i.e., materials with identical structure graphs. We then optimize hyperparameters of gradient-boosted decision trees using Bayesian optimization on the validation set and train the model using different splitting strategies, always keeping the train/validation/test ratios fixed. The figure uses the random split as the control group and computes bootstrapped mean effect sizes for the different splitting approaches. All splitting strategies lead to models with different generalization performances (better in all cases except for the density-based split) than the random control group.



Box Fig. 4 | Bootstrapped mean difference in mean absolute error for out-of-domain predictions as a function of the splitting strategy. We tuned gradient-boosted decision tree models using the different feature sets implemented in mofdscribe (currently including histograms of persistence diagrams, AMD, geometric properties, APRDF) using different featurizers on the CoRE MOF dataset to predict the methane deliverable capacity and evaluate the performance on the ARABG dataset. We ran every experiment around 20 times and then computed bootstrapped effect sizes with respect to the random split performance.

This impact of the splitting strategy also indicates the need to quantify the difficulty of a given validation split. As one method to do so, mofdscribe implements adversarial validation, which quantifies how easily a machine learning model can distinguish the train from the test set.

In the figure below, panel **a** shows the adversarial validation scores for the datasets considered in Moosavi et al.¹³². For the entries on the diagonal, we considered a random split into two equally sized parts. Scores closer to one indicate that the datasets are easily distinguishable. In this case, we see that a model can easily distinguish the datasets—in particular, the experimental ones from the in-silico assembled ones. Therefore, we cannot expect a model to necessarily generalize in this setting. In panel **b**, we see that the feature importance analysis can reveal which features the model used to distinguish the datasets. Removing those features can help to mitigate data-drift features or also help to guide the generation of new materials that can mitigate those biases (or remove materials that are dissimilar from the target distribution, i.e., have a ROC-AUC score greater than 0.5). When we group the features into scopes, as in Moosavi et al.¹³², we find that the dominating differences across databases vary. While linker feature contributions do not play a major role in distinguishing structures from the BW and CoRE databases, they do play an important role in distinguishing structures from the CoRE and ARABG databases. BW denotes a database of hypothetical MOFs assembled by Boyd and Woo¹⁴⁵ and ARABG abbreviates a database of hypothetical MOFs assembled by Anderson et al.¹⁴⁶



Benchmarks and Leaderboard

To foster the comparability of models built for digital reticular chemistry, we also implemented MOFBench classes that users can use to generate a report of the performance of their modeling pipeline on some benchmark tasks (Listing 4). The MOFBench classes ensure that all steps are performed consistently, and that different modeling strategies become comparable. They define a dataset, a splitting strategy, and a set of metrics and automatically capture the computational environment. Via a pull request on GitHub, these results can be easily added to the leaderboard that is currently part of the mofdscribe documentation (mofdscribe.readthedocs.io). Users are additionally asked to provide a file describing the modeling strategy and fill a model card,^{130,205} which will also appear in a subsection of the leaderboard (Case Study 3). We hope that mofdscribe can help pivot reticular chemistry into the digital age by giving the community tools to think and work in a data-driven manner.²⁰⁶

Case Study 3: Creating a new model and submitting it to the leaderboard

We implement a full modeling pipeline from featurization to benchmarking in the following code. Note, however, that there will be additional steps that tune features and model hyperparameters in practice.

```
from mofdscribe.featurizers import RACs, PHImage
from mofdscribe.bench import LogkHC0200DBench
from mofdscribe.featurizers.base import MOFMultipleFeaturizer
from xgboost import XGBRegressor
import pandas as pd
```

```

from mofdscribe.bench import logKHBench
# myModel must implement .fit and .predict
# myModel can contain any additional processing steps
bencher = logKHBench(myModel)
# bench() returns a pydantic model that is validated
# upon submission of a pull request
report = bencher.bench().json()

```

Listing 4: Example of the use of MOFBench. The benchmarking classes only need to be provided with a model object that implements `fit` and `predict` methods. It will then use a `Splitter` object from the `mofdscribe` package to compute cross-validated metrics on a `StructureDataset`, which are part of the report. The report also contains additional meta-information, such as the timings of different steps. It can be serialized to a JSON file that can be submitted to the leaderboard via a pull-request template in the `mofdscribe` GitHub repository.

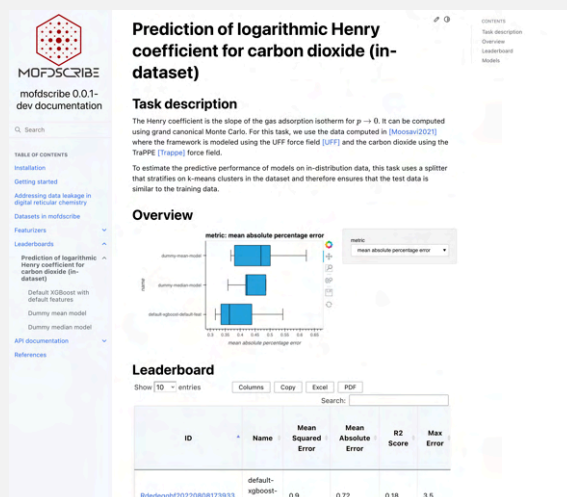
```

# We precompute all features, alternatively,
# one could also write a cached function
# that accepts Structures
featurizer = MOFMultipleFeaturizer([RACs, PHImage])
# featurize_many loops over all structures
features = featurizer.featurize_many(
    LogKHExtrapolationBench().ds.structures)
feature_frame = pd.DataFrame(data = features,
    names = featurizer.feature_labels())

# We will implement a simple XGBoost regressor
model = DFModel(XGBRegressor(), feature_frame)

# benchmark results
bencher = LogkHCO200DBench(model)
report = bencher.bench()

```



Box Fig. 6 | Exemplary screenshot of a leaderboard page.

More examples, including one on an experimental dataset, can be found in the GitHub repository (<https://github.com/kjappelbaum/>)

```
mofdscribe/tree/main/examples). The examples can be run on
Google Colab (e.g., https://colab.research.google.com/github/kjappelbaum/mofdscribe/blob/main/examples/build\_model\_using\_mofdscribe.ipynb).
```

2.3 CONCLUSIONS

While data-intensive approaches are becoming more popular in (reticular) chemistry, they are still far from being routine and standard; there are currently no standard practices for digital (reticular) chemistry.¹⁹⁵ We identified the lack of easy-to-use featurization methods and problems with model validation and comparison as the key limitations hampering the progress of the field. To address those impediments, we developed a Python package, *mofdscribe*, that provides utilities along each step from ideation to model publication. The *mofdscribe* package provides increased accessibility to machine learning for reticular chemistry and beyond without compromising rigor, especially for less experienced users. This will allow for a closer coupling of data-driven materials design and the synthesis and characterization of (in-silico generated) materials since it is very easy for non-experts to use *mofdscribe* to power machine-learning models that could be used, for instance, in an active learning workflow.²⁰⁷ While we intend to add new features to the library and maintain it, we also hope to embrace a community effort in which bugs are fixed, and the community of digital chemists and materials scientists adds new features. To facilitate this, we designed our library so that it is easy for researchers to implement new strategies, such as featurizers, in our library so that other digital chemists can easily reuse their work. We hope that, together with the open availability of machine-actionable data,²⁰⁸ our developments will systematize and accelerate machine learning for chemistry.

2.4 METHODS

2.4.1 Featurization

The details of the re-implemented featurizers are described in the original publications and the online documentation.

2.4.2 Benchmarking using automated machine learning

For the learning curves shown in Case Study 1, we trained gradient-boosted decision trees, as implemented in the XGBoost library, on the default feature set in *mofdscribe*. To mimic currently utilized settings, we ran the experiments 100 times with a random train/test split of 0.8/0.2.

For the case study analyzing the impact of splitting techniques, we used optimized gradient-boosted decision tree models, as implemented in the CatBoost²⁰⁹ library, using *optuna*²¹⁰ for a maximum of 100 trials or a timeout of 10 hours using the tree of Parzen estimators²¹¹ sampling strategy. We detail the hyperparameter grid we considered in Appendix B.6.

For Figure 13, we followed the approach from the *automatminer* library²¹² and used automated machine learning, which automatically optimizes over various models and model architectures within a certain computational budget. Concretely, we

use the TPOT library,^{213–215} which uses genetic programming for all machine learning pipeline steps, including feature engineering (for instance, using principal component analysis). We used the defaults of 100 generations with a population size of 100 but also limited the search time to 48 h and five-fold cross-validation.

2.4.3 Molecular simulations

All grand-canonical Monte-Carlo simulations for the reference dataset were performed using the RASPA code,²¹⁶ describing the force-field as a rigid framework with the UFF forcefield²¹⁷ and a cutoff of 12 Å, whereby we correct for the truncation using analytical tail-corrections.²¹⁸ Simulations were orchestrated using the AiiDA computational infrastructure.^{219,220}

DATA AVAILABILITY

Data used in this work is available via the mofdscribe package. The new dataset of predicted properties derived from grand canonical Monte Carlo simulations reported with this work is available on the MaterialsCloud²²¹ (10.24435/materialscloud:qt-cj) and has been integrated with existing data from the QMOF Database via the Materials Project’s MPContribs interface^{222,223} (10.17188/mpcontribs/1883597).

CODE AVAILABILITY

The most recent information about the tools is assembled under <https://mof.world>. The mofdscribe library is available on GitHub (<https://github.com/kjappelbaum/mofdscribe>). Documentation for the package is available on ReadTheDocs (<https://mofdscribe.readthedocs.io/>). The graph hashes are implemented in the structuregraph-helpers package, which is also available on GitHub (<https://github.com/kjappelbaum/structuregraph-helpers>). For encoding (and decoding) of elemental properties, we developed a dedicated library, element-coder, which is also available on GitHub (<https://github.com/kjappelbaum/element-coder>). The moffragmentor package is also available on GitHub (<https://github.com/kjappelbaum/moffragmentor>). The AiiDA simulation workflows are based on the ones implemented in the aida-lsmo package (<https://github.com/lsmo-epfl/aiida-lsmo>). The topological data analysis workflows are based on moleculetda, available on GitHub (<https://github.com/a1k12/moleculetda>).

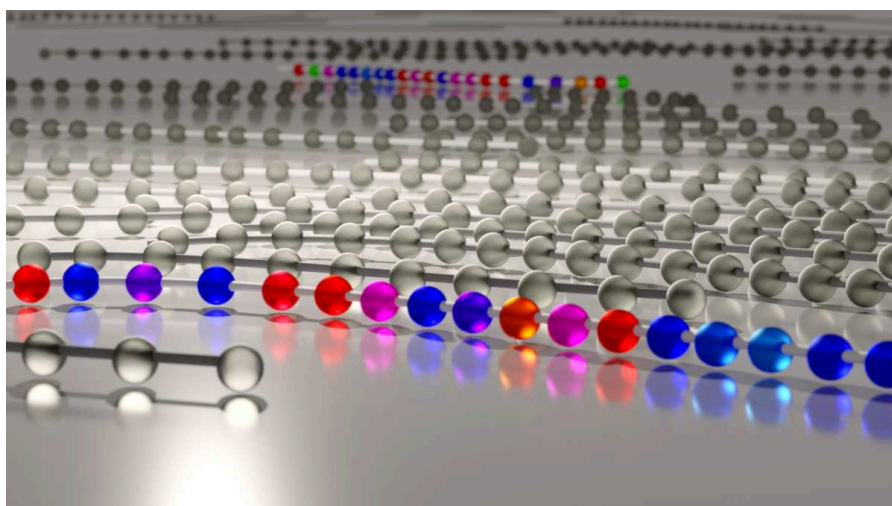
Table 2: Overview of implemented featurizers. An up-to-date list can always be found in the online documentation.

name	description
AccessibleVolume	accessible volume, computed using the zeo++ code. ^{175,176}
AMD	generalization of the average-minimum distance approach proposed by Widdowson et al. ¹⁶⁶
APRDF	generalization of the atomic-property labeled radial distribution function proposed in Fernandez et al. ²²⁴
Asphericity	shortcut for the asphericity descriptor ^{225,226} implemented in RDKit. ¹⁷¹
AtomCenteredPH	atom-centered persistent homology. Analogous to the approach reported by Jiang et al. ²²⁷
DiskLikeness	molecular descriptors computed based on principle moment of inertia, computed using RDKit. ¹⁷¹ Descriptor proposed by Wirth et al. ²²⁸ as a measure of ligand shape.

Eccentricity	shortcut for the eccentricity descriptor ^{225,229} implemented in RDKit. ¹⁷¹
EnergyGridHistogram	energy grid histograms, computed using RASPA, ²¹⁶ as proposed by Bucior et al. ²³⁰
GuestCenteredAPRDF	This featurizer builds on the APRDF featurizer, but instead of using the correlations between all atoms, it only considers the ones between the guest and all host atoms (within some cutoff distance).
Henry	Henry coefficient, as computed using RASPA. ²¹⁶
InertialShapeFactor	shortcut for the inertial shape factor descriptor ^{225,229} implemented in RDKit. ¹⁷¹
LSOP	local structure order parameters (LSOP), modified approach from R. Zimmermann and Jain ¹⁶⁵ Here we place a site at the center of mass and then compute the LSOPs around this site. In this way, we attempt to capture the shapes of full building blocks.
NConf20	molecular flexibility descriptor based on the number of energetically accessible conformers. Based on implementation of Wicker and Cooper ¹⁷³ using RDKit. ^{171,231}
NPR1	shortcut for the normalized principal moments ratio 1 ($= I_1/I_3$) descriptor ²³² in RDKit. ¹⁷¹
NPR2	shortcut for the normalized principal moments ratio 2 ($= I_2/I_3$) descriptor ²³² in RDKit. ¹⁷¹
PairwiseDistanceHist	histogram of pairwise distances between atoms in a molecule/structure
PairwiseDistanceStats	statistics of pairwise distances between atoms in a molecule/structure
PartialChargeHistogram	histogram of partial charges computed with a charge equilibration strategy (EqEq). ²³³
PartialChargeStats	statistics of partial charges computed with a charge equilibration strategy (EqEq). ²³³
PHHist	(2D) histogram of persistence diagrams, computed based on developments by Krishnapriyan et al., Krishnapriyan et al. ^{181,182} using the dionysus and diode codes (latter being a Python binding to parts of CGAL ²³⁴).
PHImage	vectorization of persistence diagrams as persistence image ²³⁵ computed based on developments by Krishnapriyan et al., Krishnapriyan et al. ^{181,182} using the dionysus and diode codes (latter being a Python binding to parts of CGAL ²³⁴).
PHStats	statistics of persistence diagrams, computed based on developments by Krishnapriyan et al., Krishnapriyan et al. ^{181,182} using the dionysus and diode codes (latter being a Python binding to parts of CGAL ²³⁴).
PHVect	vectorization of persistence diagrams using Gaussian mixture models, ^{183,184} computes using the pervect ²³⁶ library.
PMI1	first principle moment of inertia, computed with RDKit. ¹⁷¹
PMI2	second principle moment of inertia, computed with RDKit. ¹⁷¹
PMI3	third principle moment of inertia, computed with RDKit. ¹⁷¹
PoreDiameters	pore radii, computed with zeo++ ¹⁷⁶
PoreSizeDistribution	histogram of pore sizes, computed with zeo++. ¹⁷⁶ Has been used in Pinheiro et al. ¹⁷⁷
PriceLowerBound	lower bound for the MOF price based on elemental prices (a surrogate for chemistry and useful as a screening filter).
RACS	revised autocorrelation functions, as proposed by Janet and Kulik ¹⁶³ and applied to MOFs by Moosavi et al. ¹³²
RadiusOfGyration	shortcut for the radius of gyration descriptor ²²⁹ implemented in RDKit. ¹⁷¹
RayTracingHistogram	histograms of ray lengths passed through the unit cell, computed using zeo++ ¹⁷⁶ Proposed by Jones et al. ¹⁷⁸
RodLikeness	molecular descriptors computed based on principle moment of inertia, computed using RDKit. ¹⁷¹ Descriptor proposed by Wirth et al. ²²⁸ as a measure of ligand shape.
BUMatch	minimum root-mean-squared-distance between the connecting site structure of the building blocks and the “ideal” one in different nets.

SphereLikeness	molecular descriptors computed based on principle moment of inertia, computed using RDKit. ¹⁷¹ Descriptor proposed by Wirth et al. ²²⁸ as a measure of ligand shape.
SphericityIndex	shortcut for the sphericity descriptor ²²⁵ implemented in RDKit. ¹⁷¹
SurfaceArea	(probe accessible) surface areas, as computed using zeo++ ¹⁷⁶
VoxelGrid	3D voxel representations of the structure. Similar to Hung et al. ¹⁸⁹ and Cho and Lin ¹⁹⁰

3

BIAS FREE MULTIOBJECTIVE
ACTIVE LEARNING FOR
MATERIALS DESIGN AND
DISCOVERY

ABSTRACT The design rules for materials are clear for applications with a single objective. For most applications, however, there are often multiple, sometimes competing objectives where there is no single best material, and the design rules change to finding the set of Pareto optimal materials. In this work, we leverage an active learning algorithm that directly uses the Pareto dominance relation to compute the set of Pareto optimal materials with desirable accuracy. We apply our algorithm to de novo polymer design with a prohibitively large search space. Using molecular simulations, we compute key descriptors for dispersant applications and drastically reduce the number of materials that need to be evaluated to reconstruct the Pareto front with a desired confidence. This work showcases how simulation and machine learning techniques can be coupled to discover materials within a design space that would be intractable using conventional screening approaches.

CITATION This chapter is a preprint version of our article: Jablonka, K. M. et al. *Nat. Commun.* **2021**, 12.

CONTRIBUTION K.M.J developed and implemented the machine learning approach and conducted the machine learning experiments. K.M.J wrote the manuscript with B.Y. and B.S.

3.1 INTRODUCTION

The holy grail of material science is to find the optimal material for a given application. Finding the optimal material requires a metric to rank the materials. If we have a single objective, our aim is clearly defined: we evaluate the performance indicator of the materials with respect to this objective, and we can rank our materials. Developing efficient strategies to find such an optimum with a minimal number of experiments is an active area of research. In many practical applications, scientists and engineers are often faced with the challenge of simultaneously optimizing multiple objectives. Optimizing one objective alone may come at the cost of penalizing others.²³⁷ For example, in drug discovery, scientists have to balance potency or activity with toxicities and solubility. In chemical process design, engineers must optimize yields for several process units yet sacrifice energy consumption. Likewise, in material science, desirable material properties can be interdependent or even inversely related. For example, one would like a material that is both strong and ductile, and as these are inversely correlated, it is challenging to synthesize new materials that satisfy both criteria at the same time.²³⁸ In these cases, there is no unique way to rank the materials.

If one has multiple objectives, a practical solution is to combine the different performance indicators into a new overall performance indicator. But unless such an overall performance indicator is a unique, well-defined function of different performance indicators (e.g., costs), the arbitrary combination of performance parameters obscures the true nature of the optimization problem; there simply is no material that simultaneously optimizes all target properties. No single optimum is generally preferred over all the others; hence the most valuable information any search in the design space can give is the set of all possible materials for which none of the performance indicators can be improved without degrading some of the other indicators. In statistical terms, these materials are called the set of all Pareto-optimal solutions (i.e., the Pareto front). In this work, we address the question of how to efficiently search for this set of materials, and with confidence not to discard a good material. Such a methodology is particularly important if it is difficult to evaluate an unlimited number of materials because of limited resources.

Recently, there has been quite some research effort to use machine learning to design and discover new materials.^{239–243} A naive approach would be to train a machine learning (surrogate) model to make predictions for all materials in the design space and then use these predictions to compute the Pareto front. However, from a practical point of view, such an approach is inefficient, as it is unclear how to choose a training set that makes the model confident in the relevant regions of our design space. A random, or even diverse set, will probably contain more points than we need and does not consider that we do not need the same accuracy in all parts of our design space. The question we, therefore, need to answer is how we can efficiently train this model to make confident predictions in the relevant regions of our design space. An appealing way to do this is active learning.²⁴⁴ Here, we initialize a model with a small sample of our design space and then iteratively add labels, i.e., measurements or simulation results, to the training set where the model needs them most. This allows us to efficiently build a model that can solve the question of what materials are Pareto optimal and which ones we should discard for further investigation.

It is instructive to compare this approach with Bayesian optimization.^{245–250} In such an optimization, one would like to know the next best measurement by typically (and implicitly) assuming that the current evaluation will be the final evaluation.²⁵¹ Then, we can use an acquisition function to propose the next best measurement based on the predictions of a machine learning model. This best measurement can then be added to the training set and in this way, one can selectively improve the pre-

dictions of the model in a potentially promising part of the design space. However, most, if not all, of these optimization techniques rely on introducing a total order in the search space with which the materials are ranked in terms of performance. This biases the search (or introduces other technical difficulties, which we discuss in Appendix C.1). In this context, it is essential to realize that, mathematically speaking, Pareto dominance only defines a partial order in our design space. This means we can only say if a material is Pareto dominating or not, but we cannot directly compare them; hence, introducing a total order is nothing more than a (subjective) formulae on how to compare apples and pears.²⁵²

Here, we show how to recover without such bias, but with confidence, a prediction of the Pareto front in the context of polymer discovery. The rational design and discovery of polymers has been a longstanding challenge in the scientific community due to their combinatorial chemical and morphological complexity, which also requires the consideration of multiple spatiotemporal scales.^{253–255} In our approach, we use machine learning to predict the next best experiments to systematically reduce the uncertainty of our prediction of the Pareto front until all polymers within our design space can be confidently classified. To reach this goal, we use a modified implementation of the ϵ -PAL algorithm introduced by Zuluaga et al.,^{256,257} which iteratively reduces the effective design space by discarding those polymers from which we know, with confidence from our model predictions (or measurements), that they are Pareto-dominated by another polymer. To make progress in this search, we evaluate the polymer with the highest dimensionless uncertainty from the set of possible polymer candidates, which our model predicts to be near or at the Pareto optimal. The search terminates when all points are classified as Pareto efficient or discarded. Overall, this method has additional advantages that can be important for materials design and discovery applications. For example, we show how we can tune the granularity of the approximation to the Pareto front in every objective and, in this way, trade off efficiency with accuracy. Moreover, conventional active learning methods often require complete data sets, while in most practical applications, we are often faced with a situation where we have a lot of data for one property and much less for another. Our method can deal by construction, with partially missing data in the objective functions, i.e., missing one property measurement for some materials, and also can consider noise in the measurements. Therefore, given its broad applicability, we anticipate that the same workflow will accelerate the design process in the lab.

3.2 RESULTS

The polymers in our study are representative of dispersants typically used in solid suspension systems to prevent the flocculation of suspended particles, for example, to ensure the color strength of pigments in coatings applications.²⁵⁸ Finding the optimal polymer for a dispersant-based application is a typical example of a multiobjective search. One would like to obtain a polymer with optimal adhesion strength to the surface of the particles that need to be suspended. Once on the surface, the polymers need to repel the other particles, and finally, one needs to ensure that the viscosity of the solution ensures kinetic stability.²⁵⁹ Interestingly, some of these criteria are in competition with each other. For instance, we can imagine that certain monomer types will enhance both the binding to the surface and the attraction between the polymers. In this case, there is no unique solution, and we have to trade binding with the surface with the repulsion between the polymers. This is a general observation in many multiobjective problems. We often find natural, completely unavoidable, competing goals such as the strength-ductility trade-off.

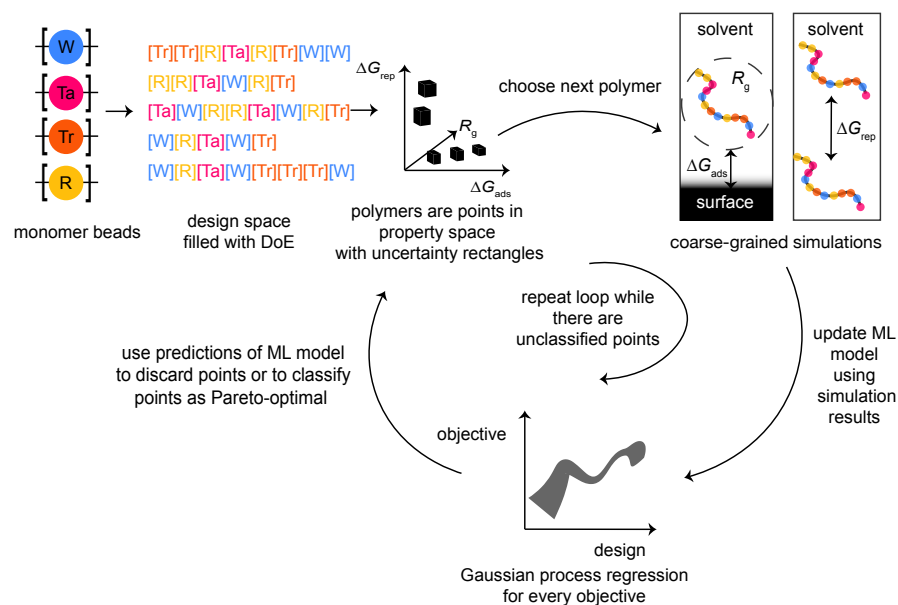


Figure 15: Overview of the workflow. Using classical design of experiments (DoE), we enumerate representative samples in the design space of monomer sequences, which we then explore in the active learning loop with the ϵ -PAL algorithm. For this algorithm, Gaussian process surrogate models provide us with predicted mean and standard deviations that enable us to decide which designs we can confidently discard, classify as Pareto-optimal, and determine which simulation we should run next to reduce the uncertainty for points near the Pareto front maximally. Models that are trained over the course of this process can reveal structure-property relationships and can be inverted using genetic algorithms to explore further the design space that has not been considered with DoE.

In this work, we mimic the design of our dispersant using a coarse-grained model (see Figure 15). Our model represents a typical linear copolymer often used as a dispersant. In this coarse-grained model, we map monomers with different interactions with the surface and the solvent to different beads, which translates to a design space containing more than 53 million possible sequences of polymer beads (see Appendix C.2). For a given hypothetical dispersant, we use molecular simulation techniques to evaluate our three (“experimental”) key performance indicators. Although we conduct the synthesis and experiments in silico, the number of possible dispersants and the required computational time to evaluate the performance is too large for a brute-force screening of all 53 million dispersants of our coarse-grained polymer genome.²⁶⁰ Therefore, also for this in silico example, we are limited by our resources, and we aim to obtain our set of Pareto optimal materials as efficiently as possible.

3.2.1 Dispersants design

The model polymers investigated in this work represent dispersants used in solid suspension systems. That is, each bead in our coarse-grained simulation represents a monomer in a copolymer (Figure 15). In practice, dispersant performance can be evaluated based on several fundamental driving forces. First, the adhesion strength of the polymer onto a suspended particle surface; second, the steric stability of the polymer, i.e., the ability to help repel suspended particles from one another; finally, the viscosity of the polymer solution, which is associated with the kinetic stability

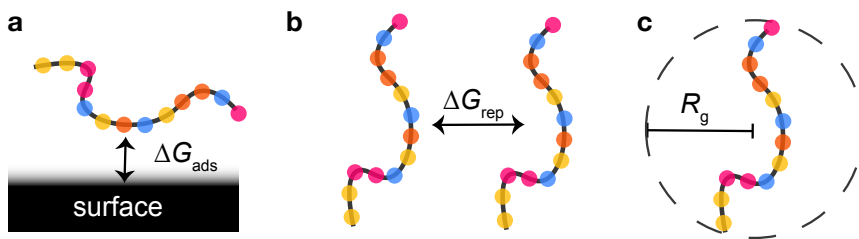


Figure 16: Schematic illustration of the polymer performance descriptors we calculate using coarse-grained simulations. **a** ΔG_{ads} is the single-molecule free energy of adsorption onto a model surface. **b** ΔG_{rep} is the dimer repulsion energy. **c** R_g is the radius of gyration, an indicator of polymer viscosity.

of the system.^{259,261} To characterize such driving forces, we calculate the following properties (Figure 16) using coarse-grained dissipative particle dynamics (DPD) simulations:²⁶² the adsorption free energy (ΔG_{ads}) onto a model surface, quantifying the adhesive strength to the surface, the dimer free energy barrier (ΔG_{rep}) between two of the same polymers, as a metric for the repulsion between the polymers, and finally the radius of gyration (R_g), a molecular property commonly associated with the polymer viscosity,^{263,264} and which can be experimentally determined using small-angle x-ray scattering.²⁶⁵

The main objective of this study is to identify polymer sequences that optimize all three of these molecular properties from a sequence design space comprised of 4 possible monomer types, with the number of monomers for each type ranging from 4 to 12.

We initially sample our polymer design space (Figure 17), i.e., the possible arrangements of monomers, by performing a full factorial experimental design on the monomer types, where each monomer type contains a selection of monomer counts. This ensures we enumerate all possible combinations of available monomer counts and types (see Methods). Compared to sampling from the latent space of generative models such as standard autoencoders or variational autoencoders, this approach maintains a high level of model interpretability. Monomer sequences are generated in random order based on these design points. We then explore the space sampled with the design of experiments using our active learning algorithm to find the Pareto-optimal polymers. An overview of our workflow is illustrated in Figure 15.

3.2.2 Pareto active learning

In this work, we are interested in not only efficiently but confidently identifying an approximation of the Pareto front. To achieve this, we need two ingredients: first, a way to discard points or to classify them as Pareto-optimal, and second, a way to propose the next best sample(s) to evaluate. Our modified version of the ϵ -PAL algorithm^{256,257} addresses these matters by using the uncertainty estimate (σ) of a Gaussian process regression surrogate model to construct hyperrectangles for a predicted material (Figure 18).

Let us assume we have two objective functions. In Figure 18, we illustrate the working principle of the ϵ -PAL algorithm. We start with a set of diverse experiments for which we measured the objectives. Based on those, we can train an initial model using features that are simple to compute and are intuitively related to the chemistry of the polymers (solely based on the monomer sequence) and can make predictions for all the polymers that are indicated as black points. For each point, we construct hyperrectangles, shown in Figure 18a, around the mean μ (which comes either from

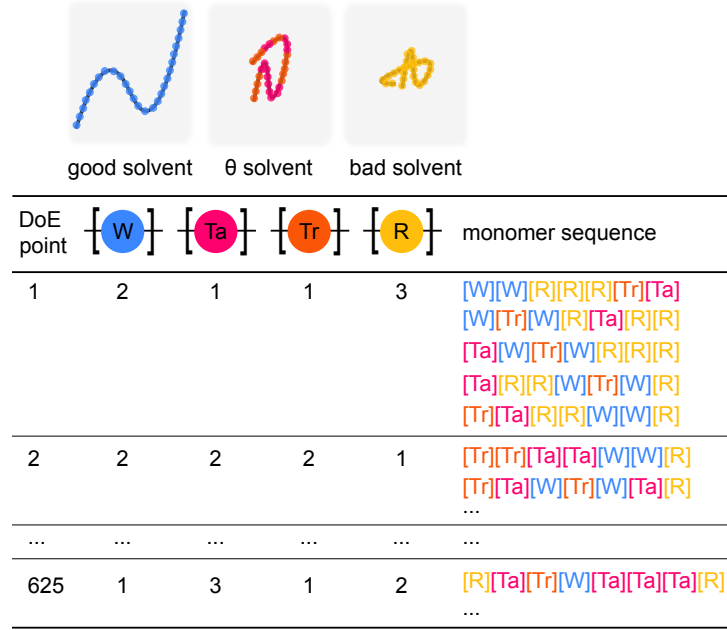


Figure 17: Illustration of the DoE approach. The beads of our coarse-grained model have different interactions with the solvent. The “[W]” bead corresponds to a polymer in a good solvent, the “[R]” bead to a polymer in a bad solvent, and the “[Ta]”, and “[Tr]” beads to polymers in a theta solvent. “[Tr]”, and “[Ta]” differ from each other in their interaction with the surface. For each DoE point, which specifies the composition of a polymer, we sample five arrangements of monomers. This results in a design space of 3125 polymers in total. Note that the polymers we sampled had at minimum 4 units of each monomer.

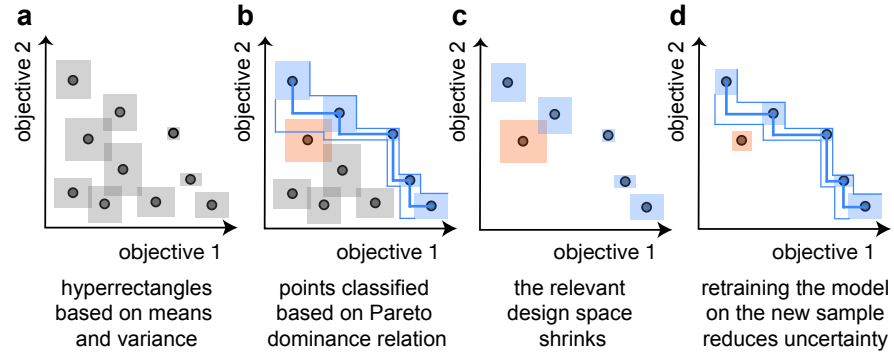


Figure 18: Illustration of the working principle of the ϵ -PAL algorithm. **a** For each point, we construct hyperrectangles around the mean μ (coming from either from the model predictions or the measurement) with widths proportional to the uncertainty σ (which is the standard deviation of the posterior of the points we did not sample yet and the estimated uncertainty of the measurement for the sampled points; the exact width of the uncertainty hyperrectangles is also a function of the hyperparameters and the iteration). **b** Using the ϵ -Pareto dominance relation, we can identify which points can be discarded with confidence and which are with high probability Pareto optimal. **c** After this classification, the design space that is relevant for the search is smaller, and we can sample the largest hyperrectangle to reduce uncertainty (the orange one in this case). **d** After performing the simulations for the sampled material (orange), the model uncertainties decrease, notably in the neighborhood region of the sampled material.

the model predictions or the measurement) with a width that is proportional to the uncertainty σ (the standard deviation of the posterior) for the points we did not sample so far, and the estimated uncertainty of the measurement for the sampled points (the exact width of the uncertainty hyperrectangles is also a function of the hyperparameters and the iteration, see appendix for details). The lower and upper limits of these hyperrectangles are the respective pessimistic and optimistic predicted performance estimates for all the objectives.

From the (ϵ) -Pareto dominance relation, we can identify those points that can be discarded with confidence (gray in Figure 18b) and those of which are with high probability Pareto optimal (colored blue) as shown in Figure 18b. If the pessimistic estimate for our predicted material is greater than a tolerance (defined using the ϵ hyperparameter) above the optimistic estimate for all other materials, it will be part of the Pareto front. Our current estimate of the Pareto front is then the (thick) blue line connecting the blue points. In addition, we can make a simple estimate of the accuracy of our current prediction of the Pareto front by connecting the bottom left corners of hyperrectangles associated with our current estimate of the front, which gives us the most pessimistic front (lower blue line). The optimistic front is then obtained by connecting the upper right corners. For the case of multiobjective maximization using this algorithm, we can discard materials with high certainty if the optimistic estimate of the material is within some set tolerance (ϵ) below the pessimistic estimate of any other material. We maintain the orange point as it cannot be discarded within our set uncertainty, see Figure 18b. Hence, we have a simple geometric construction that allows us to classify whether a predicted material is Pareto-optimal or whether we can discard it with certainty.

After this classification, we can with certainty discard all experiments in which the hyperrectangles are completely below the most pessimistic front. This significantly reduces our design space. In terms of Bayesian optimization, this can be thought of as the exploitation step.

Following this classification, the next step is determining the next material to run experiments on. The next material to characterize should be the one that reduces our uncertainty in classifying points as Pareto optimal. For this, we assume that the uncertainties are normalized by the predicted mean such that the area of our hyperrectangles represents the relative error (i.e., we use the coefficient of variation). We then simply improve the information gain of our model the most if we reduce the uncertainty of the largest rectangle among points presumed near or at Pareto front. In Figure 18c, the biggest area corresponds to the orange point, and adding an extra point will improve the accuracy of our model in that part of the Pareto front. As a result, we obtain a more accurate estimate, see Figure 18d. We can continue this procedure by sampling the next largest hyperrectangle until our prediction of the Pareto front has reached the desired accuracy. The model is then retrained using all sampled points, including those that have been discarded.

It is interesting to note that all points we discard are with high probability not part of the ϵ -Pareto front. Hence, we do not need to sample points from this region of design space even though those points may contain the largest uncertainty regions out of the entire set. Interestingly, by choosing the hyperparameters properly, we can also obtain theoretical guarantees on the quality of the Pareto front. That is, given a kernel of a predictive Gaussian process model and proper scaling parameters of the hyperrectangles, ϵ will be the maximum error of our Pareto front with probability δ (see Appendix C.10).²⁵⁶ Setting a larger tolerance ϵ will speed up the classification of the design space but increase the errors. In practice, it is reasonable to set ϵ to be larger than the error of the experiment/simulation.

Here, we use this algorithm to efficiently choose which simulations to run, although, in principle, one can apply the same algorithm to efficiently choose experiments—

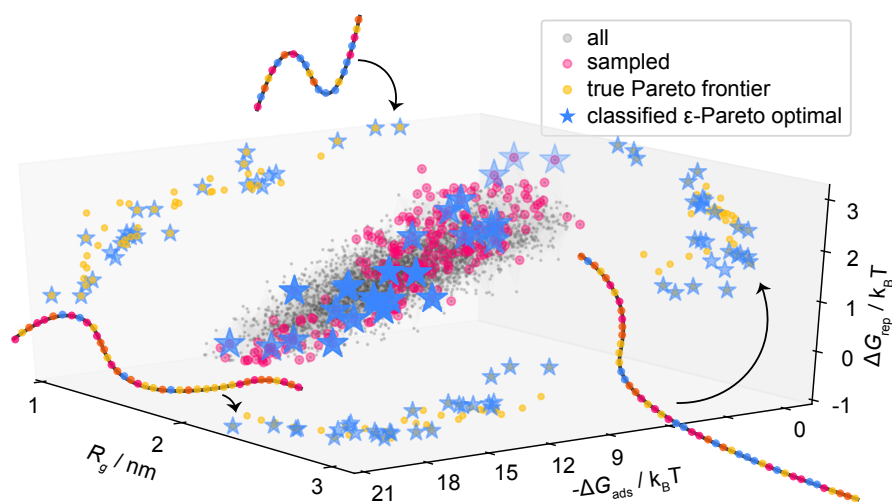


Figure 19: Representation of polymers in property space. Simulations have been performed on the entire experimental design to determine the three key performance indicators, the adsorption free energy (ΔG_{ads}), the dimer free energy barrier (ΔG_{rep}), and the radius of gyration (R_g). Each gray point corresponds to the performance of a unique polymer. Points sampled or classified as ϵ -Pareto optimal by the ϵ -PAL algorithm are marked in magenta and blue, respectively. Pareto optimal points have also been projected on their respective 2-D planes. The schematic drawings of the polymers indicate that the Pareto optimal materials in our design space have vastly different compositions, e.g., showing a large difference in the degree of polymerization. (See Appendix C.8.)

for example, in self-driving laboratories,²⁶⁶ or in other related multiobjective materials discovery problems where we want to recover the Pareto front within some level of granularity ϵ .

For this study, we have performed brute-force simulations and obtained property estimates for all design points generated from our DoE approach to evaluate the algorithm's effectiveness. This allows us to recover the true Pareto front (in the space sampled with DoE) and compare it to our predicted Pareto front obtained after each active learning cycle. Figure 19 presents the property estimates, Pareto optimal points, and the sampled points in property space.

A key metric for evaluating the quality of the Pareto front is the so-called hypervolume indicator. This indicator measures the size of the space enclosed by the Pareto front and a user-defined reference point (in 2D, this would equate to the enclosed area) and is commonly used to benchmark Bayesian optimization algorithms. In general, a better design will always have a larger hypervolume.²⁵² Using this indicator, we analyze how accurately and rapidly our active learning approach recovers the true Pareto front. Additionally, we compare our approach with random sampling. Note that random sampling might seem like a naive approach; however, it has been shown to be an efficient search method, for example, for outperforming grid search in many optimization problems.²⁶⁷ Hence, it is a relevant baseline.

Figure 20a illustrates the working principle and effectiveness of the algorithm. It attempts to classify the polymers in the design space as fast as possible into either an ϵ -accurate Pareto optimal or a discarded polymer. Each iteration corresponds to the (in silico) synthesis of a new dispersant and subsequent evaluation of the three key performance indicators, the adsorption free energy (ΔG_{ads}), the dimer free energy barrier (ΔG_{rep}), and the radius of gyration (R_g). The data show that already after 10 iterations, the algorithm confidently discards many polymers (orange region) and

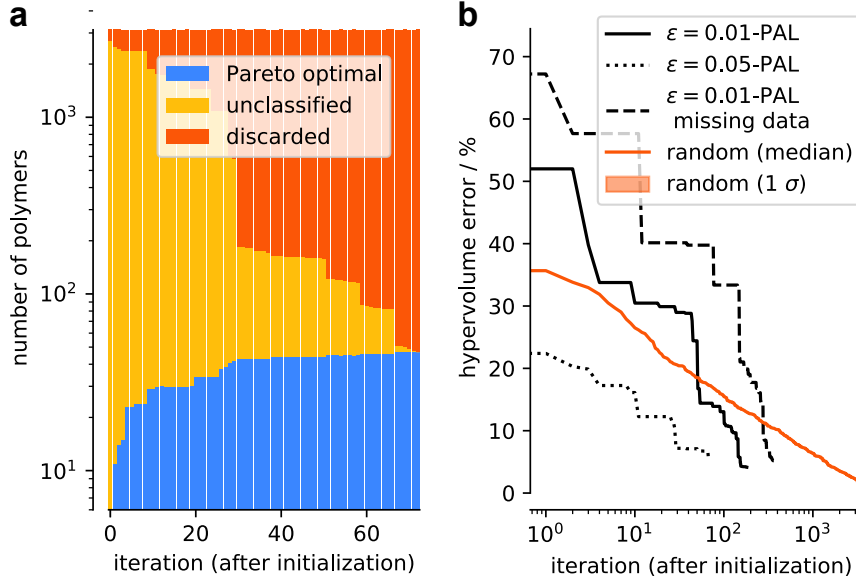


Figure 20: Classified points and hypervolume error as a function of the number of iterations. **a** The ϵ -PAL algorithm classifies polymers after each learning iteration with $\epsilon_i = 0.05$ for every target and a coregionalized Gaussian process surrogate model. The Gaussian process model was initialized with 60 samples selected using a greedy farthest point algorithm within feature space. Note that the y -axis is on a log scale. **b** Hypervolume errors are determined as a function of iteration using the ϵ -PAL algorithm with $\epsilon_i = 0.01$, and 0.1 for every target. A larger ϵ_i makes the algorithm much more efficient but degrades the final performance. For $\epsilon_i = 0.01$, we intentionally leave out a third of the simulation results for ΔG_{rep} from the entire data set. The method for obtaining improved predictions for missing measurements with coregionalized Gaussian process models is discussed in more detail in Appendix C.7. Hypervolume error for random search with mean and standard deviation error bands (bootstrapped with 100 random runs) is shown for comparison. For the ϵ -PAL algorithm, we only consider the points that have been classified as ϵ -accurate Pareto optimal in the calculation of the hypervolume (i.e., with small ϵ the number of points in this set will be small in the first iterations, which can lead to larger hypervolume errors). All search procedures were initialized using the same initial points but varied substantially after only one iteration step due to the different hyperparameter values for ϵ . Note that the x -axis is on a log scale. Overall, missing data increases the number of iterations needed to classify all materials in the design space.

finds many ϵ -accurate Pareto optimal polymers (blue region). In Figure 20b, we compare the algorithm's performance with random search and use the hypervolume error—the relative error with respect to the maximum hypervolume of the design space—to quantify the quality of the estimated Pareto front. We can observe that ϵ -PAL achieves the target error ($\epsilon = 0.01$) with more than 89 % fewer iterations compared to the random exploration of the design space (153 with our approach, 1421 with random search).

An extension of our approach is a case where we have missing data. Often in experimental data sets, data is missing for a more difficult-to-measure property. In our case, calculating the dimer repulsion energy requires significantly more computational time than the other properties. Hence, it would be interesting to see how such an algorithm performs if, say, 30 % of the data is lacking for one of the properties (i.e., one of the properties is immeasurable for some of these materials). Figure 20b presents the algorithm's performance when a third of the dimer repulsion energies are missing. In this situation, using independent Gaussian processes for each objective and running a subsequent experiment with a missing datum would not improve model predictions for that property. The idea is that we capture correlations between our various objectives by means of coregionalized Gaussian process models.²⁶⁸ These models allow us to predict multiple objectives using a single surrogate Gaussian process model and provide better estimates for missing objectives if one (or more) of the objectives is missing while all others are present for a given design point (see Methods).

3.2.3 Chemical insights

Interestingly, we can not only use our surrogate models as part of the design loop to expedite the discovery process, but we can also obtain some understanding of structure-property relationships.

We use the SHAP technique to obtain chemical insights into what the models learned during the discovery process. This method can reveal how the features used by the model influence the predictions and how those features interact with one another.²⁶⁹ In our case, we use SHAP to understand structure-property relationships. In Figure 21, we list the five features that, according to our machine learning model, are most important for every target in order of relevance.

Let us first focus on the radius of gyration (Figure 21a). The most important feature for the prediction of the radius of gyration is the degree of polymerization, followed by the number of good solvent segments ($[W]$), the number of bad solvent segments ($[R]$), the number of theta solvent beads ($[Tr]$), and the relative entropy of the monomer sequence. From Flory's scaling relation, we know that the radius of gyration scales with the chain length N : $R_g \sim N^\nu$, where ν is the Flory exponent.²⁷⁰ We find that our model detects this direct proportionality between chain length and the radius of gyration. This showcases that our model captures theoretically consistent relationships during the active learning process. More interestingly, we can see that the SHAP analysis on the last two features already highlights a critical difference between the theory and our model: Our model provides us with insights into what happens when we change the composition (e.g., increase the ratio of $[W]$ or $[R]$). For example, if we increase the fraction of good solvent beads ($[W]$), we have a higher radius of gyration, while the radius decreases if we increase the number of bad solvent beads ($[R]$). Hence, our model closely recovers our intuition and lets us quantitatively capture these effects.

We use the same machine learning model to predict the two other key performance parameters, the interaction with the surface and repulsion between the dispersants, and also use SHAP to extract the feature importance. Here we see that by

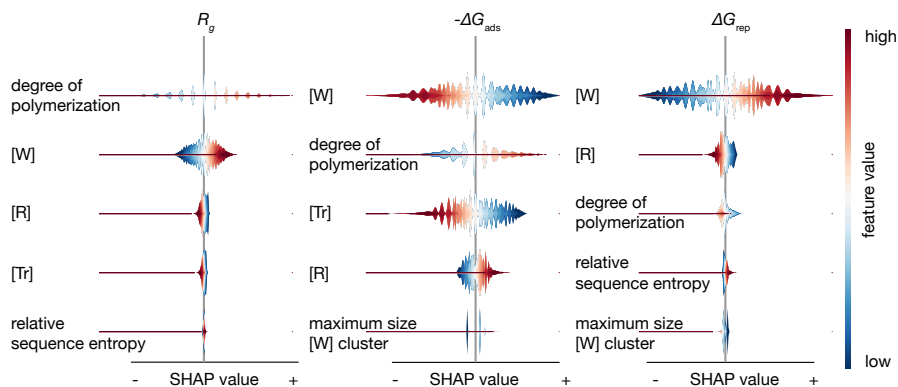


Figure 21: Influence of the feature values on the predictions. SHAP summary plots for the models for our three objectives based on the models obtained after running the ϵ -PAL algorithm in Figure 20. We used all the sampled points from the run of the ϵ -PAL algorithm as background data for the SHAP analysis. Red points correspond to a high feature value, whereas blue points correspond to low feature values. The width of the violin shown on the x -axis corresponds to the density of the distribution of SHAP values and indicates how the features impact the model output. A negative SHAP value means that the specific feature value decreases the predicted value with respect to the baseline prediction. SHAP values were computed for a coregionalized model with a Matérn-5/2 kernel and $\epsilon = 0.05$.

increasing the ratio of [W], we decrease the interaction with the surface but increase the repulsion between the polymers. Interestingly, we find that for the dimer repulsion energy, increasing the relative sequence entropy of the monomers increases the repulsion between dimers. This implies that if one plans to maximize the repulsion between the polymers, one should increase the disorder of the arrangement of the monomers, i.e., avoid blocks. Importantly, we also see that the feature relevance varies between targets, highlighting why a multiobjective search—in contrast to an independent single-objective search—is pertinent when aiming to accelerate the multiobjective materials discovery process.

3.2.4 Inverse design

To investigate whether our algorithm missed potentially better-performing polymers that we did not consider in our experimental design, we inverted the machine learning models that were trained on-the-fly during the active learning cycle. To do so, we use elitist genetic algorithms (GAs) to find novel polymer structures that maximize the output of our models while biasing the generation of polymers to ones that are different from the monomer sequences that we considered in the DoE (by adding explicit novelty terms into the loss function, see Appendix C.11). This exploits our machine learning model's ability to capture relevant regularities from the design space.

Figure 22 shows the property distribution of the best-performing polymers we found based on the output of the GA compared to our original results. We find that independent of whether we bias the GA towards exploration or exploitation, we cannot find polymers that Pareto-dominate the points that we found using our combination of the DoE and ϵ -PAL approaches.

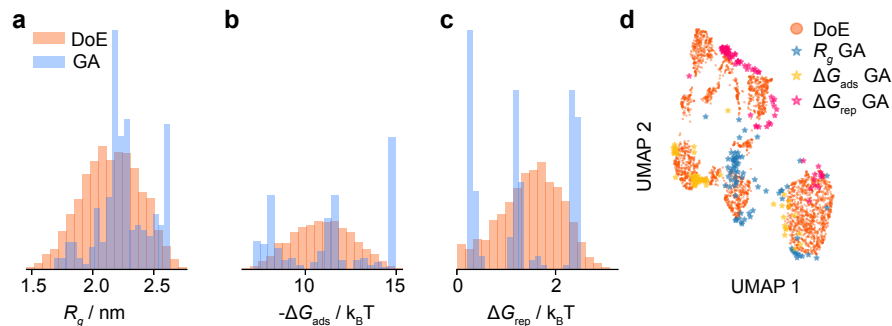


Figure 22: Distribution of properties for the polymers found by inverting the machine learning models using a GA. To generate new feature vectors, we ran the genetic algorithm with different weights for the novelty part of the loss function—ranging from no penalty for similar polymers to high (50 times the objective) penalty for polymers similar to the ones already sampled with our DoE. Additionally, we run the genetic algorithm for different elitist ratios. For each feature vector, 3 possible polymer bead sequences were generated since a feature vector does not map to one unique arrangement of monomers. **a** Distribution of radius of gyration (R_g). **b** Distribution of adsorption energies ΔG_{ads} . **c** Distribution of dimer repulsion energies ΔG_{rep} . **d** Polymer properties obtained from GA are projected onto the first two uniform manifold approximation and projection for dimension reduction (UMAP) components and compared to those obtained from DoE.

3.3 CONCLUSIONS

In materials design, one typically has to balance different objectives, and the proper weighting of these objectives is usually unclear in the early design stages. This insight raises the need for a method to efficiently identify the Pareto optimal points while not discarding interesting materials. Using key thermodynamic descriptors derived from molecular simulations for a large polymer design space, we show that our materials design approach can be used to explore polymer genomes that would be intractable using conventional screening methods. Our approach finds the relevant polymers in a fraction of the evaluations that are needed using traditional approaches and provides us with predictive models and structure-property relationships on-the-fly while being robust to missing data. This showcases how the coupling between data-driven and conventional materials design approaches, such as simulations or experiments, can greatly enhance the rate with which we discover or optimize materials while concurrently giving us insights into structure-property relationships.

The vision behind our approach is that in a multiobjective optimization, the only rigorous result one can obtain is the set of Pareto optimal materials. Hence, one should focus on an algorithm that systematically improves the accuracy of the estimated Pareto front. Ranking the materials in a multiobjective optimization introduces, by definition, a bias, and detailed studies have been made to identify how such bias can impact the optimization (see Wagner et al.²⁷¹ and Appendix C.1. However, one can make a bias-free ranking of the experiments that improve the accuracy of the Pareto front the most. This observation can be translated into an ϵ -PAL machine learning algorithm, and our case study shows that significant gains in efficiency can be achieved.

As multiobjective optimization is such a general problem, we expect that this approach can be adapted to those cases in which efficiency is essential.

3.4 METHODS

3.4.1 Coarse-grained model

In our model, the polymer bead diameters are assumed to be greater than the Kuhn length, i.e., polymers follow the ideal chain behavior. There are four different polymer bead types in addition to one solvent and two surface bead types. Each bead type [W] - “weakly attractive”, [R] - “repulsive”, [Ta] - “theta attractive”, and [Tr] - “theta repulsive” was created based on their solvent, [S], interaction. Bead types [Ta] and [Tr] are representative of beads for homopolymers in a theta solvent but differ based on their attractive or repulsive interaction with the surface monolayer bead type [S2]. Bead type [R] is the most adsorptive on our model surface, whereas bead type [W] is the least attractive. More details on the interaction parameters are provided in Appendix C.4.

3.4.2 Design of experiments

The first step in the workflow involves the generation of polymers based on our experimental design space. To effectively sample from this design space, we used a full factorial experimental design with the number of factors equal to the number of bead types (4), the number of levels equal to the number of possible bead count variations (5 possible: 4, 6, 8, 10, 12) and 5 unique monomer sequences for each point. While this is certainly not representative of the entire sequence design space of our polymers, we assume that sequence effects come secondary to monomer content for this work. This assumption is a reasonable approximation, as noted in Appendix C.3. Overall, we obtained 3125 unique linear polymer molecules represented by their monomer sequence. The experimental design was created using pyDOE.²⁷²

3.4.3 Simulation Protocol

All simulations were set up using Enhanced Monte Carlo (EMC) version 9.4.4^{273,274} and run with Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) version 2018/03/16.²⁷⁵ Using EMC, monomer sequences for coarse-grained polymers were directly ported into LAMMPS input files.

Free energy calculations

Free energy calculations were performed using the LAMMPS plugin in Software Suite for Advanced General Ensemble Simulations (SSAGES) version 0.82.²⁷⁶ Steered molecular dynamics (SMD) simulations²⁷⁷ were first performed to generate initial configurations for each polymer-surface or polymer-polymer center of mass separation distances of a given polymer.²⁶² Umbrella sampling was subsequently performed and analyzed using the weighted histogram analysis method (WHAM)²⁷⁸ to estimate both the adsorption free energies of the dispersants onto the model dispersion surface and the dimer free energies of dispersants.

3.4.4 Machine learning

Featurization

We calculated features such as the degree of polymerization, the relative sequence entropy, the nature of the end groups (one-hot encoded), summed interaction parameters, and the nature of clusters based on the monomer sequence.

All features were z-score standardized using the mean and standard deviation of the training set (using the scikit-learn Python package²⁷⁹). More details can be found in Appendix C.6.

Gaussian process regression surrogate models

Intrinsic coregionalization model (ICM) Gaussian process regression models²⁶⁸ (of rank 1) were built using the GPy Python library²⁸⁰ based on Matérn-5/2 kernels. In Appendix C.7 we show that coregionalization improves the predictive performance in the low-data regime, i.e., the initial setting of the algorithm. The ICM models assume that the outputs are scaled samples from the same Gaussian process regression (GPR) (rank 1) or weighted sum of n latent functions (rank n). A higher rank is connected to more hyperparameters and typically makes the model more difficult to optimize. We provide a performance comparison of rank 1 and rank 2 models in Appendix C.7. Hyperparameter optimization was performed with random restarts and in regular intervals as training points were added. More details can be found in Appendix C.7.

Feature importance analysis

We used the SHAP technique marginalized over the full DoE dataset (summarized with weighted $k = 40$ means-clustering) to calculate model interpretations²⁶⁹ and the full DoE dataset to calculate SHAP values. For the GPR surrogate models, we apply the “KernelExplainer” method. Model interpretations for runs with different ϵ are qualitatively consistent; the plot shown in the main text was computed for $\epsilon = 0.05$ and a coregionalized model with Matérn-5/2 kernel.

Pareto active learning

We implemented a modified version ϵ -PAL algorithm²⁵⁶ in our Python package, PyePAL. Our algorithm differs from the original ϵ -PAL algorithm by using the coefficient of variation as the uncertainty measure rather than the predicted standard deviations. Moreover, our implementation does not assume that the ranges (r_i) of the objectives are known. This is, instead using $\epsilon_i \cdot r_i$ for the computation of the hyperrectangles, we use $\epsilon_i \cdot |\mu_i|$ (see Appendix C.10). PyePAL generalizes to an arbitrary number of dimensions as opposed to the original MATLAB code provided by Vivek Nair²⁸¹ (limited to 2), and by default sets the uncertainty of labeled points to the experimental uncertainty or the modeled uncertainty. In addition to supporting standard and coregionalized Gaussian processes surrogate models, our library interfaces with other popular modeling techniques with uncertainty quantification, such as quantile regression and neural tangent kernels. It also offers native support for missing data, for example, when using coregionalized Gaussian processes, support for both single point (as done in this work) and batch sampling, and the option to exclude high variance points from the classification stage.

Implementation details and hyperparameter settings in this work are provided in Appendix C.10. Initial design points used to train the zeroth iteration model were selected using greedy farthest point sampling in feature space.²⁰¹ Hypervolume errors shown in the main text were calculated using the nadir point as our reference point.

Our code makes use of the following Python packages: GPy,²⁸⁰ jupyter,²⁸² lightgbm,²⁸³ matplotlib,²⁸⁴ neural-tangent,²⁸⁵ nevergrad,²⁸⁶ numba,²⁸⁷ numpy,²⁸⁸ pandas,²⁸⁹ scipy,²⁹⁰ scikit-learn.²⁷⁹

Inverting the GPR models

To invert the GPR model, we trained gradient boosted decision tree (GBDT) surrogate models with a reduced feature set (e.g., dropping the relative entropy of the monomer sequence) on the predictions of the GPR models. An elitist GA²⁹¹ was then used to maximize the output of the model while being penalized for creating invalid polymer features, i.e., features that cannot be converted to a valid monomer sequence using a backtracking algorithm, or features that are very similar to those already present in our dataset. More details can be found in Appendix C.11.

3.5 DATA AVAILABILITY

The input files for the molecular simulations and the analysis results of the simulations are available on the Materials Cloud²²¹ Archive (DOI: 10.24435/materialscloud:8m-6d).

3.6 CODE AVAILABILITY

Code for the machine learning part (including the featurization and genetic algorithm) of this study is available as part of the `dispersant_screener` Python package (archived on Zenodo: 10.5281/zenodo.4256868, and developed on GitHub: github.com/byooooo/dispersant_screening_PAL). A general-purpose implementation of the ϵ -PAL algorithm that can be used with other models such as quantile regression, is available as the `PyePAL` package (archived on Zenodo: 10.5281/zenodo.4209470, and developed on GitHub: github.com/kjappelbaum/pyepal).

Part II

APPLICATIONS

USING COLLECTIVE KNOWLEDGE TO ASSIGN OXIDATION STATES



ABSTRACT Knowledge of the oxidation state of metal centers in compounds and materials helps understand their chemical bonding and properties. Chemists have developed theories to predict oxidation states based on electron-counting rules, but these can fail to describe oxidation states in extended crystalline systems such as MOFs. Here we propose using a machine-learning model, trained on assignments by chemists encoded in the chemical names in the CSD, to automatically assign oxidation states to the metal ions in MOFs. Our approach only considers the immediate local environment around a metal center. We show that it is robust to experimental uncertainties such as incorrect protonation, unbound solvents, or changes in bond length. This method gives good accuracy, and we show that it can be used to detect incorrect assignments in the CSD, illustrating how machine learning can capture collective knowledge and convert it into a useful tool.

CITATION This chapter is a preprint version of our article: Jablonka, K. M. et al. *Nat. Chem.* **2021**, *13*, 771–777.

CONTRIBUTION K.M.J designed the machine learning approach, performed the experiments, and wrote the article with B.S.

4.1 INTRODUCTION

Oxidation states are a useful concept for understanding the properties and reactivity of materials.²² Their history goes back to the early days of chemistry when Lavoisier coined the word oxidation and Wöhler the expression “oxydationsstufe” (old German spelling for the term oxidation number).^{292,293} Oxidation states are central to balance redox reactions,²⁹⁴ for chemical nomenclature,²⁹⁵ and above all to help chemists to systematize and reason about (redox) reactivity and spectroscopic properties.^{296–298} For example, Mn(II) corresponds to a formal electron configuration of $3d^5$, Mn(IV) to $3d^3$, and Mn(VII) $3d^0$. Chemists know that permanganate (Mn(VII)) compounds are highly oxidizing and often of purple color, whereas Mn(II) compounds are less reactive and typically colorless. From the oxidation state assignment in MnO_4^- we also directly know that the 3d electrons of manganese are strongly interacting with the oxygens (we formally assign them to oxygen).

Importantly, oxidation states are not quantum mechanical observables but rather a “convenient fiction” that helps chemists think about chemistry.²⁹⁹ The IUPAC defines oxidation states as “...the charge of this atom after ionic approximation of its heteronuclear bonds ...”.^{300,301} This definition is, however, too generic and cannot be readily translated into a recipe to determine the oxidation state of any given compound.

For crystalline materials, the oxidation state is often estimated using the bond valence sum method.³⁰² This method, which dates back to Linus Pauling,³⁰³ approximates all bonds as fully ionic, and the oxidation state is estimated by summing up all bond valence terms S_{ij} , which are calculated based on a parametrization of an exponential function of metal-ligand bond lengths (R_{ij}):

$$S_{ij} = \exp\left(\frac{R_0 - R_{ij}}{b}\right), \quad (2)$$

where R_0 and b are empirical parameters. This technique has also found application in estimating the valence of non-metals.³⁰² There is an ongoing effort in tuning the bond valence sum method to automatically evaluate the entries in the CSD,^{304–306} which is the largest collection of metal-organic crystals.

However, the bond valence sum method is far from ideal as it has some ambiguities. First, one needs to assign bonds between the atoms, for which many different algorithms have been proposed given that the definition of a bond can be debated.^{307–309} Second, many different parameter sets exist which are derived for different systems, e.g., metal oxides or metal-organic complexes.³⁰² Yet, these parameters might need to be mixed to cover chemical space³⁰² when certain parameters are unavailable in one parametrization.

Finally, the functional form for the bond valence method might sometimes be too rigid as it is based solely on bond lengths. Technically, this can cause problems when one wants to apply the same tools on experimental structures and DFT-relaxed ones or when the bond lengths are not very accurately determined. This approach is also different from how chemists intuitively think about oxidation states, which often is more closely related to the shape of the coordination polyhedra. For example, chemists will intuitively associate a linear copper complex with Cu(I), whereas they will associate a tetragonal complex with Cu(II).³¹⁰ This intuition chemists have is built on the fact that the electron configuration can be related to the coordination geometry via concepts like the ligand-field theory.

In this context, it is important to note that quantum chemical calculations are of limited use. From a fundamental point of view, one might think that state-of-the-art quantum chemical calculations would give us the total energy for the different oxidation states, and hence it would be straightforward to determine the oxidation state

that gives the lowest energy. Unfortunately, for most MOFs the unit cell is so large that one has to use DFT, which tends to favor compounds with lower d orbital occupancy and leads to non-integer oxidation states for multivalent compounds, such as 2.5 for the irons in magnetite (Fe_3O_4), due to the self-interaction error (in the generalized gradient approximation (GGA)).^{311,312} Other computational techniques have been developed based on charge-partitioning schemes, but as the charge on an embedded atom is ill-defined and subject to charge-transfer interactions with the ligands,^{313,314} also these methods are unable to remove the ambiguity in the assignment of oxidation states. Hence, in practice, quantum calculations require a “guess” of the oxidation state as input rather than giving us insights into the oxidation state. Fundamentally, this is due to the fact that all methods to derive oxidation states, be it computational or spectroscopic, need to introduce some additional rules to assign oxidation states, for example, via comparison to reference systems or by specifying how to count electrons based on wave functions as oxidation states are not quantum mechanical observables.³¹⁵

In summary, there is a need for a new approach toward the assignment of integer oxidation states that can capture the intricacies of chemistry—to provide starting points for DFT calculations and to support chemical reasoning.

In this work, we propose to use the collective knowledge of chemists to assign oxidation states, replacing the rule-based deductive approach of formal counting rules with a fully inductive one. The approach described here harvests the collective (noisy) knowledge of chemists to create a consensus assignment of oxidation states, which to our knowledge has not been explored to provide a simple solution to this important practical question.

Towards this goal, we parsed the chemical names in the CSD for oxidation states of metal centers, numerically encoded the local chemical environment, and trained an ensemble of ML models, which makes predictions based on a “vote” between four base models, to classify the oxidation state. We chose to focus on MOFs as their experimental structures are archetypal examples for many of the reasons automatic deductive techniques might fail to assign oxidation states: Unbound solvent molecules are present in many experimental MOF structures, and sometimes the structures also contain charge compensating counter ions. Moreover, the model is challenged by problems like missing or incorrect protonation and atomic disorders. Even if our main focus in this work is on predicting the oxidation state of metal centers in MOFs, we also demonstrate that the model that was trained only on MOFs can transfer to other types of materials, such as binary ionic solids, or simple metal complexes.

4.2 RESULTS AND DISCUSSION

To create our data set of oxidation states for metal centers in MOFs, we leveraged the fact that the chemical names of nearly half of all entries in the CSD¹⁵⁰ contain oxidation states in parentheses following the metal names (as it is recommended in the guidelines for inorganic nomenclature, the IUPAC red book).²⁹⁵ This assignment, manually curated by the editors of the CSD, can be based on different arguments: chemical intuition, founded on knowledge of the chemical literature and experience with similar reactions and compounds, some computing protocol (e.g., the bond valence method), or spectroscopic evidence. Even if these oxidation states are not assigned with a unique and well-defined protocol, several chemists (at least the authors and the editor at the CSD) consider this assignment to be correct.^{150,316} The central assumption in this work is that some individual assignments might be wrong, but if enough chemists work on similar systems, the collective knowledge will be

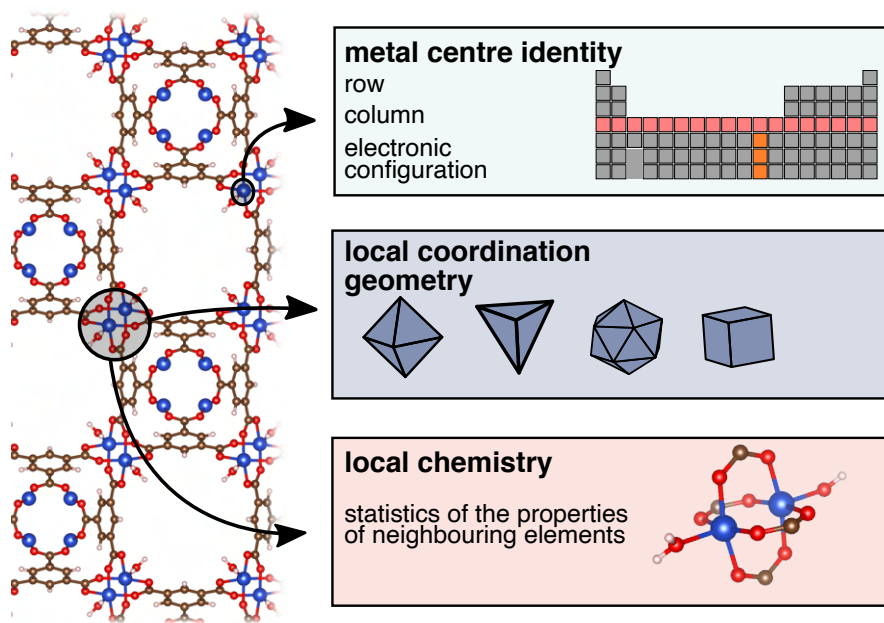


Figure 23: Schematic representation of the featurization approach. Schematic representation of the featurization approach. The feature vector takes into account three aspects. The metal type is considered through its position in the periodic table (row and column). The geometry of the coordination environment of the metal center is captured by measuring the similarity of the actual coordination environment to ideal coordination environments—this reflects chemical knowledge such as “square-planar Pt is usually d^8 ”. Four polyhedra are shown as examples. The chemical environment of the metal center is based on statistics of the elemental properties of a metal and its nearest neighbors; for example, electronegativity difference, which has an influence on how many electrons a neighboring atom is likely to donate. An illustrative metal paddlewheel cluster is shown (atom color code: Cu, blue; O, red; C, brown; H, white).

right. Similar to image recognition, the task of assigning oxidation states can be trivial to some chemists, but teaching computers how to do it opens completely new avenues due to the scalability. Many works in the past have focused on leveraging information from the CSD, but most focused on directly using the structures, for example, to build statistics of bond lengths and angles for conformational analysis,³¹⁷ leading the structure correlation principles founded by Bürgi and Dunitz.³¹⁸ In this work, we combine the structure information with another piece of information, the name of a chemical compound, that researchers typically do not associate with the CSD.

4.2.1 Encoding local environments and machine learning

In this work, we use a ML model to capture the knowledge of chemists about the oxidation state. To be able to train ML models, one has to encode the local environment as a vector of numerical descriptors (“features”). This is commonly known as featurization, and the success of any ML model crucially depends on selecting features that are able to describe the problem at hand—ideally in a physically meaningful way.^{239,319} We based our featurization approach on the locality approximation in which we consider only the immediate local environment around a metal center in a structure (cf. Figure 23 for an illustration). This is also reflected in Pauling’s principle of local charge neutrality³²⁰ and the nearsightedness principle of electronic matter,

which describes the density change caused by a potential change far away is small.¹⁶¹ In addition to being physically meaningful, this approximation allows us to create a large training set that enables powerful similarity-based reasoning. This realization reflects Pauli's parsimony principle, which states that the number of unique local environments is limited. Using the locality approximation, we can also consider structures with unbound solvent molecules and missing or incorrect protonation, as those solvent molecules or missing protons are typically outside the local environment of a metal center.

Our feature vector combines the three aspects chemists have identified as key to the oxidation state: the metal type, the geometry of the coordination environment, and the chemical environment (see Figure 23). Importantly, this featurization is based on chemical insights—but gives the flexibility to accommodate cases in which classical rules fail.^{321,322} We used the first two values of the feature vector to identify the position of the metal in the periodic table, i.e., its row and group number. The column encodes the well-known principle that elements in the same group share similar chemistry, and the addition of the row makes the encoding of the metal position in the periodic table unique. We further added the number of electrons in the different shells of the neutral atom as additional features for the metal center to additionally encode the range of possible oxidation states.

The next elements of our feature vector recognize that there is a deep relationship between coordination geometry and the electronic configuration. Prime examples for this relationship, which have been used in the past to propose an extension of the bond-valence sum method,^{323–326} are the ligand field splittings for different coordination environments, the Jahn-Teller distortion for degenerate electronic configurations, or the valence-shell electron-pair repulsion (VSEPR) model.^{327,328} To encode these effects numerically, we use order parameters, which measure the similarity of the coordination environment to a collection of ideal coordination environments (e.g., octahedral, tetrahedral, bent linear).^{309,329} In this way, we capture heuristics like “square-planar Pt is usually d^8 ” which experienced chemists can rely on but which are difficult to comprehensively encode in a deductive approach. Importantly, our featurization does not explicitly depend on bond lengths, which makes it more robust and interoperable—e.g., we can use the same model on DFT optimized and experimental structures.

The key insight on which formal counting rules are built is that different ligands are thought to donate a different number of electrons.³³⁰ We attempted to encode this more flexibly by calculating statistics, like the electronegativity differences, of elemental properties between the metal center and its geometrical nearest neighbors.³³¹

The matrix describing the immediate local environments was then used as an input for a voting classifier which arrives at its final prediction by averaging the predictions (probability of oxidation states) of four base models, each based on a different approach (extremely randomized decision trees, boosted decision trees, nearest neighbor, and linear functions) to make the estimates maximally uncorrelated, similar to chemists that use different ways of reasoning about oxidation states. This voting makes our predictions more robust and gives us an uncertainty estimate.³³² This approach is similar to the way in which we use the collective knowledge of chemists at the level of the training data to arrive at a data-driven definition for the oxidation state—not all chemists use the same method to assign the oxidation state but taken together the collective assignment for a particular chemical environment can be robust.

After calculating the feature vector for each metal site, we split the data into disjoint sets for training and testing (see Methods for more details). In addition to that, we also use structures with strong spectroscopic evidence for the oxidation state

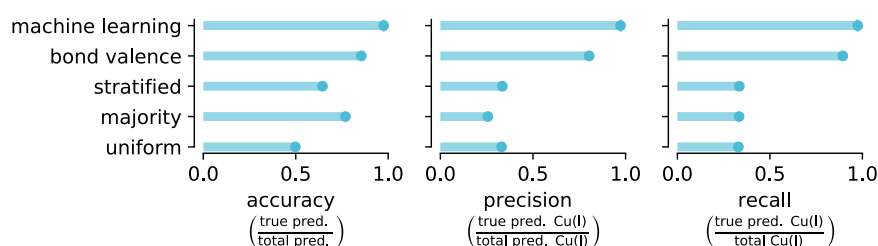


Figure 24: Performance metrics. Accuracy, precision, and recall for the assignment of oxidation states for Cu in the MOF subset of the CSD using different classifiers. The barplots show accuracy (i.e., what fraction of the predictions is correct), recall (e.g., how many Cu(II) we identify from all Cu(II)), and precision (e.g., how many Cu(II) we identified are actually Cu(II)). “Uniform” represents the same probability for both oxidation states. “Majority”: all structures are assigned the majority class, here II. “Stratified”: stratified sampling was used for the training set distribution—for example, assigning Cu(II) with a 75.8 % chance (see Table 23 for the frequency of oxidation states in the MOF subset of the CSD). Bond valence: oxidation states assigned using the bond valence sum method. Machine learning: oxidation states assigned using the approach described in the present study. In the equations, total prediction refers to how often a given oxidation state was predicted in our dataset; the true predictions are the subset of those for which the prediction was correct.

assignment as separate test cases.

4.2.2 Performance assessment

To assess the accuracy of our method, we first focused on copper, for which we can compare our results with an optimized and validated bond valence sum method.³⁰⁴ In addition, for copper both oxidation states I and II are well represented in the MOF subset of the CSD (Cu(I): 24.2 %, Cu(II): 75.8 %).

To determine the performance of our ML method and the bond valence method, we calculate the accuracy (i.e., what fraction of our predictions is correct) of our predictions as well as measures for sensitivity (recall, e.g., how many Cu(II) we identify from all Cu(II)) and precision (e.g., how many Cu(II) we identified are actually Cu(II)). These metrics are important as due to the imbalanced distributions of Cu(I) and Cu(II), we already have a 75 % chance of success by assuming all oxidation states to be II (majority vote in Figure 24). In addition to the metrics calculated for the bond valence sum method and the presented model, we also report the performance for random assignment of oxidation states (random sampling) and random sampling with a sampling probability of Cu(I/II) equal to the frequency of Cu(I/II) in our dataset (stratified sampling, i.e., assigning Cu(II) with a ca. 75 % chance) as baselines. These baselines are important for a fair evaluation of a classifier, as even a random sampling might achieve high accuracy on an imbalanced dataset. Figure 24 clearly shows that the machine-learning model outperforms the baselines and the bond valence method in all metrics.

It is interesting to use our ML results to investigate why the bond valence method fails for some structures. For this, we projected our feature space onto two dimensions using PCA (cf. Figure 25), a statistical technique that attempts to capture most of the variance in only a few principal components (here two). In these principal components, the two most relevant feature values are the extent to which the copper is trigonal co-planar with coordination number three and the extent to which copper is square co-planar with coordination number four. The black stars are structures for

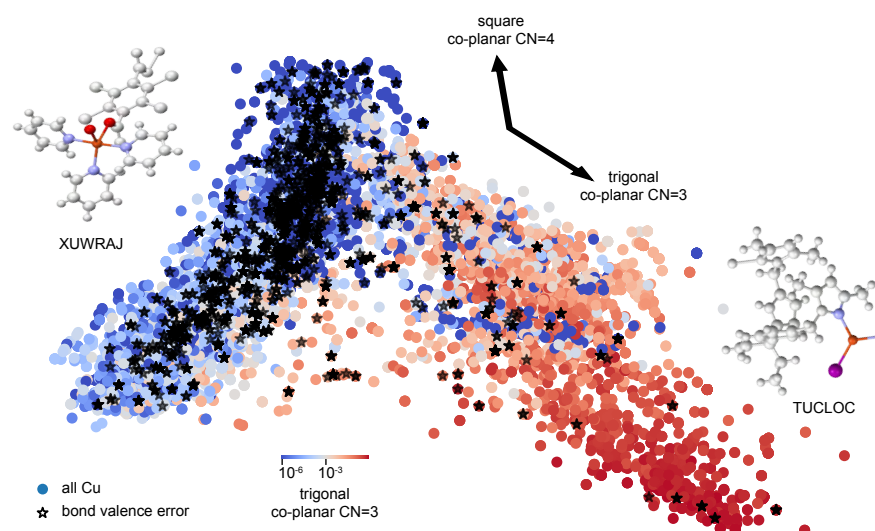


Figure 25: First two principal components for Cu sites. Projection of the feature space onto the two first principal components (linear combinations of features that capture most of the variance in the data). Every material is a dot in this figure. The arrows show the direction of the two features “square co-planar CN = 4”, and “trigonal co-planar CN = 3” that have the highest loadings in the space spanned by the first two principal components. The color coding shows the value of the order parameter of the trigonal planar coordination (logarithmically scaled color map in which low values of the “trigonal planar” order parameter are shown in blue, and high ones—indicating high similarity to “trigonal planar” coordination—in red). Black stars mark metal sites of structures for which the bond valence method predicted the wrong oxidation state. We also show two structures (CSD refcodes XUWRAJ, a coordination polymer with Cu(II) center and tetrachloroterephthalate-based ligand, and TUCLOC, a Cu(I) coordination polymer with dipyrrolylmethane-base ligand) that are at the extremes of the first principal component and for which the bond valence sum method is wrong and correct, respectively. For these structures, we show the first coordination sphere in color, where copper is orange, oxygen blue, nitrogen blue, and iodine violet). The figure shows that the errors cluster for low values of the trigonal co-planar feature.

which the bond valence method predicts the oxidation state incorrectly. We can see that these incorrect assignments cluster for copper with low values for the trigonal co-planar order parameter. In the presented machine-learning model, we see that for these structures, the geometric features are of higher importance, and exactly these geometric features can not explicitly be described in the distance-based bond valence approach.

By design, our method is directly applicable to all metals but will be less accurate for metals and oxidation states that are less frequent in the current training set. To obtain a more detailed measure of the success of our predictions, we used a test set of 42,463 metal sites that were not used in the training set to compute the confusion matrices that tabulate the predicted against the true oxidation states, for different parts of the periodic table (cf. Figure 26a). All oxidation states were correctly assigned for the easy s block (e.g., Li, Na, Ca), which generally only adopts either the +I or +II oxidation state. Even for the more challenging d block (e.g., Fe, Cu), where we have a wider range of accessible oxidation states, p block (e.g., Al, Pb, Bi), and f block (e.g., Ce, Eu, Ho), we obtained success rates of at least 90 %. These results translate

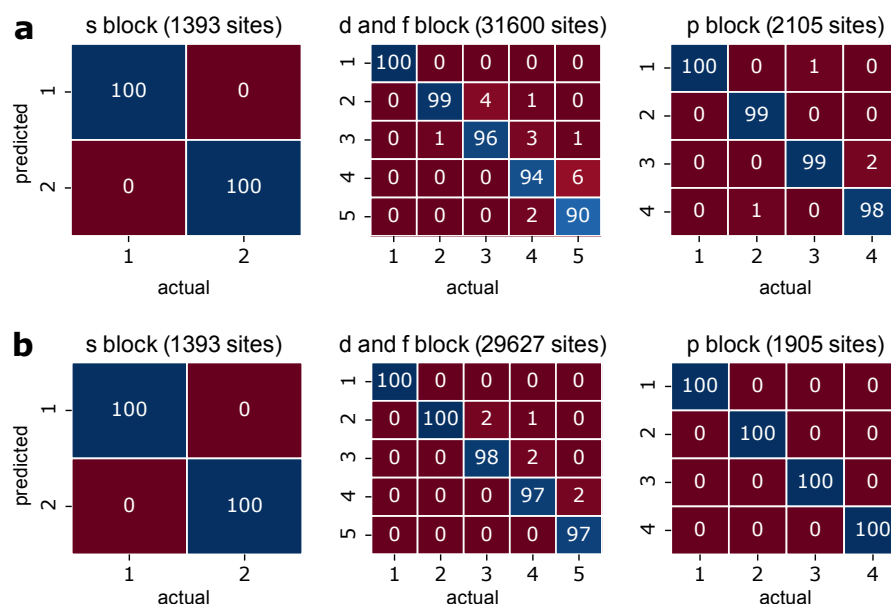


Figure 26: Predictive performance across the periodic table. Confusion matrices compare the actual and predicted assignments. In the ideal case, the off-diagonal elements are all zero. The numbers in the matrices are percentages and are normalized with respect to the columns (i.e., the percentages of actual oxidation state numbers add up to 100 %). The percentages are also used for the color coding. **a**, Confusion matrices for all predictions, independent of the uncertainty of the model. **b**, Confusion matrices only for predictions for which all four base estimators in our ensemble agree (39,943 sites, s block: 100 %, d and f block: 94 %, p block: 90 %). Confusion matrices were calculated for predictions on a holdout test set of 42,463 metal sites. We did not remove all typos listed in Appendix D.2 for this analysis.

in commonly used classification metrics that exceed 96 % (see Appendix D.2).

One additional advantage of ensemble ML models is that they can estimate how reliable a prediction is. Models based on different hypothesis spaces tend to disagree when used outside the domain of applicability (i.e., when they extrapolate) and agree when the queried case is well represented in the training data. For the machine-learning approach described here, we find a mean difference between the number of disagreeing base estimators (i.e., members of the ensemble model) of 0.89, which indicates that usually one base estimator will disagree in the case of a wrong prediction. If we use this to eliminate predictions in which the model is uncertain, we find that the overall prediction accuracy increases significantly. We now also get near-perfect predictions for the p, low valence d, and f block metals (see Figure 26b).

It is instructive to investigate those cases in which we make a prediction with high confidence yet make a wrong assignment. These structures (ca. 300) were flagged, and we retrieved the article to manually inspect the oxidation state. Out of these, in 70 cases, we observed that the assignment in the CSD did not match the one in the original paper (see Appendix D.2), sometimes caused simply by the exchange of IV to VI or I with II. In the rest of the articles, we question the assignment of the oxidation state for several of them, and, of course, we also have cases in which our method incorrectly assigns the oxidation state. All these cases are listed in Appendix D.6. The fact that many of the cases with discrepancies are erroneous assignments in the CSD suggests that it would be advantageous to use our method as a diagnostic; if we make a high-confidence prediction that differs, a more detailed investigation into

the oxidation state would be advisable.

To further confirm the accuracy of our predictions, we identified a number of structures for which the oxidation state assignment is supported by strong spectroscopic evidence. Also here, the model showed a good performance by predicting the correct results in all but one of over 50 cases, including mixed-valence cases (cf. Table 38).

The one structure for which our method failed is a Ce(IV)-MOF. As there are not many structures with Ce(IV) in the MOF subset of the CSD, we suspected this to be a case where our predictions are limited by the coverage of Ce(IV) in the training set. Therefore, we extended our training set to all structures in the CSD (i.e., MOFs and non-MOFs, see Appendix D.8), and for this training set, the present machine-learning approach correctly predicts Ce(IV).

4.2.3 Case studies

It is interesting to look at the assignments of a few case studies in detail. Of particular interest are MOFs with mixed-valence and the case of flexible MOFs for which there is considerable discussion in the literature about the oxidation state.

The importance of geometrical features in assigning the oxidation state is evident for the case of the mixed-valence MOF Cu(I/II)-1,3,5-tricarboxylate (BTC).³³³ Mixed-valence MOFs have been excluded from our training set as the CSD does not systematically indicate which oxidation state corresponds to which metal. As our features are local, we can use the model to determine the oxidation state for each metal site in these mixed-valence MOFs. Since our program does not consider the symmetry, we determine the oxidation state for each of the 16 metal sites in the Cu(I/II)-BTC unit cell separately (cf. Figure 27). In agreement with the experimental data³³³ we assign the four coppers in the paddlewheel (Figure 27c) to be +II while the eight coppers in the macrocycle (copper carboxylate ring shown in Figure 27a) are assigned to be +I. In Figure 27, we schematically illustrate the relative importance of the different features that determine the assignment. To estimate the feature importance, we use the SHAP technique that is built based on a game-theoretic concept in which one wants to estimate the fair payout for a player in a game.^{269,334} In our case we want to determine the “fair” contribution of the features to the prediction and, in essence, analyze the change in the prediction of the model for all possible combinations of features without the feature of interest. A large positive SHAP value indicates that the feature increases the predicted value. From this analysis, we see that the assignment is mostly based on the local coordination geometry (blue ring in the pie charts in Figure 27), where for the paddlewheel the square pyramidal (sq. pyr.) and for the macrocyclus the linear order parameter is the most important feature. In Figure 27d we give the top five features that determine the oxidation state. In these figures, each dot corresponds to one of the 16 metal sites, and the color corresponds to the value of the features. If the structure were perfectly symmetric, there would be only two dots. The order parameters for the coordination environment reflect that the Cu in the paddlewheel is considerably square pyramidal (high values for sq. pyr.) but not linear, while the opposite is true for the Cu in the macrocyclus. This nicely illustrates how the machine-learning approach described here captures the chemical intuition that a square pyramidal coordination environment is always associated with Cu(II), whereas a linear coordination environment is associated with Cu(I).

An interesting case of a flexible MOF is MIL-47. For this MOF Barthelet et al.³³⁵ reported an oxidation of the V(III) center upon desorption of a terephthalate guest molecule, which also resulted in a change of flexibility of the framework. In contrast, Centrone et al.³³⁶ found no evidence for such a change in oxidation state. The model presented here supports the initial assignment, as also did follow-up studies.^{337,338}

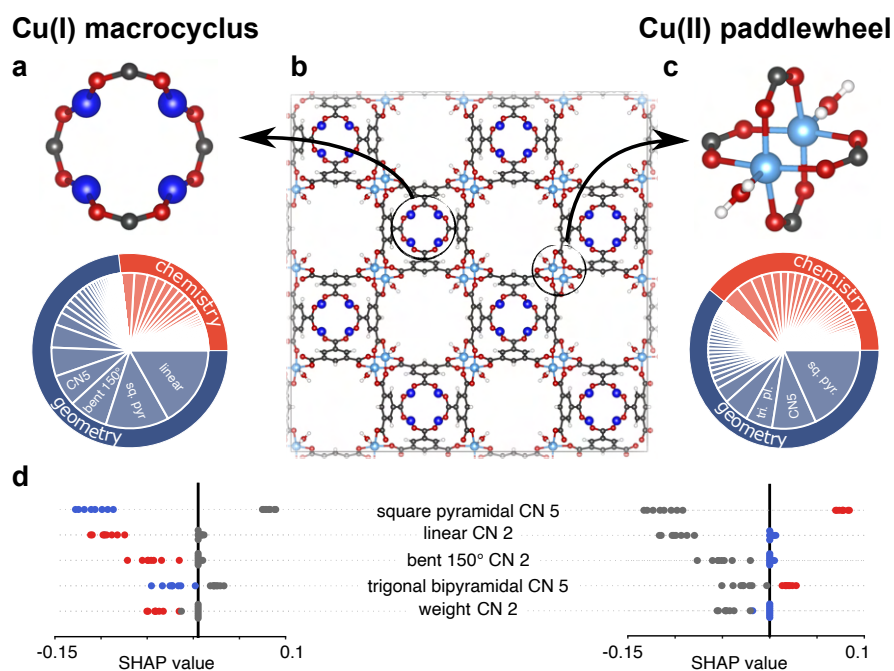


Figure 27: Predictions of the oxidation states in mixed-valence MOF Cu(I/II)-BTC. **a**, Cu(I) carboxylate macrocycle, with below a schematic representation of the relative importance of the geometrical features compared with the chemical ones for the assignment of the Cu(I) sites (CN: coordination number, sq. pyr: square pyramidal,). **c** Crystal structure of Cu(I/II)-BTC. **c**, Structure of the Cu(II) paddlewheel; here, too, the geometrical features are of high importance for the assignment of the Cu(II) sites (tri. pl: trigonal planar). Atom color code: Cu(I), blueberry blue; Cu(II), light blue; O, red; C, brown; H, white. **d**, Summary of the Shapely additive explanations (SHAP)²⁶⁹ for the Cu(I) and Cu(II) sites. Red shows a feature is of high value, and blue shows a feature is of low value. Cu(II) values are gray on the SHAP value plot for Cu(I), and vice versa. A negative SHAP value (shown on the abscissa) translates into a lower predicted oxidation state, whereas a positive SHAP value corresponds to a higher oxidation state.

For the crystal structure with the terephthalate guest molecule (cf. Figure 28a) we find vanadium in the oxidation state +III, whereas we find vanadium in the oxidation state +IV for the crystal structures without the guest molecule (cf. Figure 28d). As visible in Figure 28, the structures show a subtle change in the coordination geometry upon activation, which the model mostly captured in a change of the order parameter for octahedral coordination (coordination number 6 order parameters in Figure 28e) which is higher for the structure with the guest molecule. This reflects the chemical intuition that V(III) is regularly octahedrally coordinated and that the regular octahedron is distorted upon oxidation.³³⁹ It is difficult to capture such effects in deductive approaches like formal electron counting or with the functional form of the bond valence method.

Another peculiar example of the importance of small geometrical details in the assignment of the oxidation states is a redox-active MOF of the MOF-74 type in which the iron center was shown to be oxidized upon O₂ adsorption at room temperature, which was also reflected in a slight change in the coordination geometry of O₂ from end-on (η^1) to a rather side-on (η^2) coordination.³⁴⁰ The model presented here can recognize the change in oxidation state based on the slight change in coordination geometry. In a classical bond valence or ligand-counting analysis, the assignment would remain ambiguous due to the dependence on the arbitrary choice

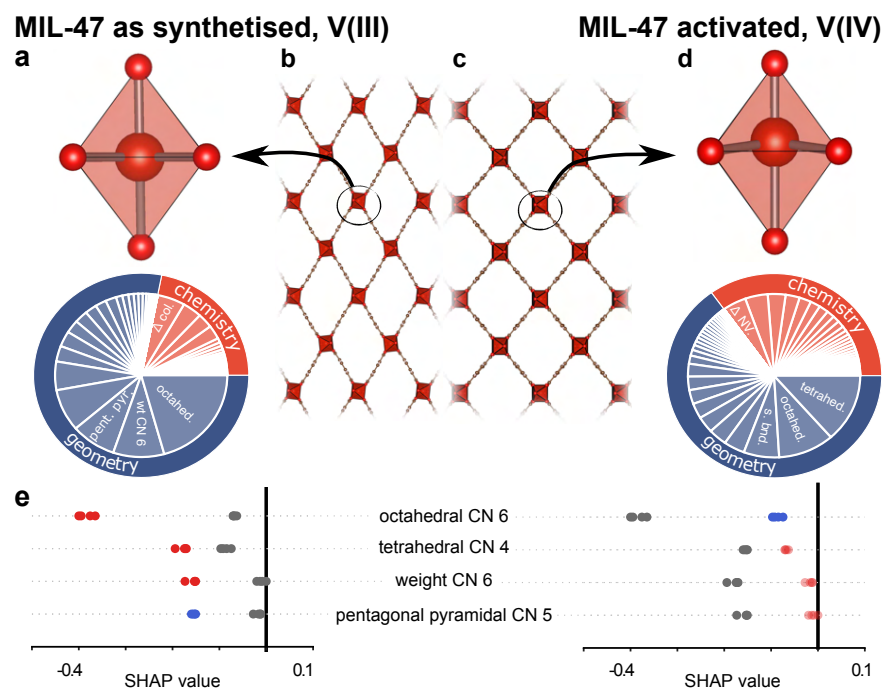


Figure 28: Predictions of the oxidation states in MIL-47 before (as-synthesised) and after activation. **a**, Octahedral coordination in the as-synthesized structure. **b** Crystal structure of the as-synthesized structure, with below a schematic representation of the relative importance of the geometrical features compared with the chemical ones for the assignment of the oxidation states of the vanadium sites (CN: coordination number, pent. pyr: pentagonal pyramidal, Δcol difference in the column number). **c** Crystal structure of the activated structure. **d** Distortion of the octahedron after activation. Here, too, the geometrical features are of high importance for the assignment of oxidation states (s. bnd: single bond, ΔNV : difference in the number of valence electrons). Atom color code: V, orange; O, red; C, brown; H, white. **e**, Summary of the Shapely additive explanations (SHAP)²⁶⁹ for the V(III) and **f** V(IV) sites. Red shows a feature is of high value, and blue shows a feature is of low value. V(III) values are shown in gray on the SHAP value plot for V(IV), and vice versa. A negative SHAP value translates into a lower predicted oxidation state, whereas a positive SHAP value corresponds to a higher oxidation state.

of the method to assign bonds between the atoms.

4.2.4 Novel MOFs

An interesting question is how well we would predict the oxidation state of a novel MOF of which the chemistry is different from structures currently in the CSD. One way to estimate the transferability of the model is to test it on databases of other structure classes, including binary ionic solids from the Materials Project²²², transition metal complexes³⁴¹ and COFs,¹⁴⁷ without additional training. For small transition metal complexes, chemists conventionally assign oxidation states by adding up the electron donation of ligands around the metal center. For ionic crystals, on the other hand, chemists will usually base their reasoning directly on the chemical formula. The model presented here unifies this picture: for the cases in which the model is highly confident in its prediction, we can predict the oxidation state with almost the same accuracy as for MOFs (see Appendix D.5.5–D.5.6). Moreover, from

a more practical point of view, these results give confidence that the approach presented here will predict reasonable oxidation states for novel classes of MOFs that are not yet in the CSD.

We provide an app that uses our pre-trained model to assign oxidation states of metal centers of MOFs on the Materials Cloud. This app requires the crystal structure as input and outputs the oxidation states of the different metal sites together with an estimate of the confidence. In addition, the program can provide details on the feature's importance.

4.3 CONCLUSIONS

Oxidation states are a fundamental concept in chemistry. For many compounds (like simple metal complexes), we can write down the oxidation state from empirical knowledge. The bond valence sum method is successful in assigning oxidation states of more complex structures. For small systems, we can even carry out accurate quantum calculations to estimate the oxidation state.³⁴² However, there are many structures for which these approaches are of limited use. Yet, chemists have provided a large amount of data on the oxidation state of structures for which these conventional approaches cannot be used. In this work, we show that with an appropriate set of descriptors, this collective knowledge can be converted into a surprisingly powerful tool. Our work highlights the power data-driven techniques can have in chemistry and materials science; as an example to solve fuzzy problems where no reliable alternative exists but relying on the collective knowledge acquired by chemists.

4.4 METHODS

We used the CSD Python API to retrieve the chemical names for the structures of the MOF subset of May, 2019.¹¹⁶ Regular expressions were used to parse the oxidation states and the corresponding metals. We excluded 6921 structures from our modeling workflow due to atomic overlaps in the experimental structure.

For featurization, we used the `matminer` Python package¹⁵⁹ and standardized (based on standard deviation and mean of each column of the training set) all features prior to use in the modeling process.

The ML model adopted in this work is a soft voting classifier using gradient boosting, *k*-nearest neighbors, logistic regression, and an extra trees base classifier implemented in the `scikit-learn` library.²⁷⁹ For hyperparameter optimization of each base estimator, we used a mixed strategy of random search, simulated annealing, and the tree Parzen estimator (tpe) algorithm using the `hyperopt-sklearn` library to avoid biases due to a single search strategy. Classification probabilities were calibrated on a validation set, disjoint from training and test set. We use soft voting to be able to provide an uncertainty metric. Further, this approach is appealing as it gives higher weight to more confident models. More details can be found in Appendix D.2.

To ensure that test errors are not optimistically biased due to multiple similar, but not identical, local environments in one structure, we not only constrained the split into training and test set to have the same ratios of oxidation states and elements (iterative stratification³⁴³) but also to include all chemical environments of one structure in only one set. That is, if one chemical environment of a structure appears in the training set, all other chemical environments of the same structure will not appear in the test set. Identical fingerprints are automatically discarded from our

training set. We perform this split based on “base identifiers” of the CSD database identifiers, which we create by stripping all trailing integers. This accounts for the fact that some entries in the CSD are updated entries (e.g., with refined lattice constants) for the same structure for which a trailing number has been added to the original identifier. By restricting all structures with the same base identifier to be in the same set, we avoid data leakage.

Further, we use a submodular selection approach³⁴⁴ to select a smaller, diverse set of training points to make our training more efficient (and again recognize the parsimony principle of Pauling by minimizing redundancy in our training set). To address the fact that some metals (like copper) are more than an order of magnitude more frequent than other metals (like ruthenium), we adjusted our sampling procedure to randomly subsample the structures with the most common metals (Cu, Zn, Cd). Crystal structures were drawn using VESTA.³⁴⁵

DATA AVAILABILITY

The feature matrices, labels, and a pre-trained model are deposited on the Materials Cloud archive (DOI: 10.24435/materialscloud:2019.0085). Data and code that reproduces the plots shown in the main text can be found in a Code Ocean capsule (DOI: 10.24433/CO.3636895.v1).

CODE AVAILABILITY

Predictions for MOF structures can be performed using the `oximachinerunner` Python package (<https://github.com/kjappelbaum/oximachinerunner>), which is installable from PyPI. The code for parsing, featurization as well for the ML models is available on GitHub (https://github.com/kjappelbaum/learn_mof_ox_state/tree/master, https://github.com/kjappelbaum/oximachine_featurizer) and deposited on Zenodo (DOIs: 10.5281/zenodo.3567011, 10.5281/zenodo.3567274). The web app is hosted on the work section of Materials Cloud (go.epfl.ch/oximachine).²²¹ The code for this app, along with a Dockerfile, is also available on GitHub (<https://github.com/kjappelbaum/oximachinetool>) and deposited on Zenodo (DOI: 10.5281/zenodo.3603606). The code used to generate the graphical abstract is available in ref. [346].

5

MACHINE LEARNING FOR INDUSTRIAL PROCESSES: FORECASTING AMINE EMISSIONS FROM A CARBON CAPTURE PLANT



ABSTRACT One of the main environmental impacts of amine-based carbon capture processes is the emission of the solvent into the atmosphere. To understand how these emissions are affected by the intermittent operation of a power plant, we performed stress tests on a plant operating with a mixture of two amines, 2-amino-2-methyl-1-propanol, and piperazine (CESAR1). To forecast the emissions and model the impact of interventions, we developed a machine-learning model. Our model showed that some interventions have opposite effects on the emissions of the components of the solvent. Thus, mitigation strategies required for capture plants operating on a single component solvent (e.g., MEA, monoethanolamine) need to be reconsidered if operated using a mixture of amines. Amine emissions from a solvent-based carbon capture plant are an example of a process that is too complex to be described by conventional process models. We, therefore, expect that our approach can be more generally applied.

CITATION This chapter is a preprint version of our article: Jablonka, K. M. et al. *Sci. Adv.* **2023**, *9*, eadc9576.

CONTRIBUTION K.M.J designed the machine learning approach, performed the machine learning experiments, and wrote the article with B.S. and feedback from the other authors.

5.1 INTRODUCTION

The most well-known and broadly used benchmark solvent to capture CO_2 is monoethanolamine (MEA).¹¹ Energy efficiency, however, is not the only criterion that is important in selecting a solvent for a carbon capture process. Amine emissions are equally important, as these may require cost-incurring gas treatment strategies to meet the operational permits and address environmental concerns.^{348,349} At present, we do not have a clear understanding of such amine emissions from a capture plant operating with these new solvent mixtures such as CESAR1.^{350,351}

Amine emissions from carbon capture plants are one example of an industrial process for which the plant's design, control, and optimization require detailed knowledge of how the process parameters interact and impact the operation of the plant and what the (chemical) mechanisms and rate constants are. Due to the complexity of such plants, process models typically focus on capturing the steady-state operation.³⁵² But there are many cases where operation beyond the steady state is required. For instance, the design and operation of current and future power plants will need to constantly adapt to the increased share of intermittent renewable energy generation.^{353–356} This requires tools that fully capture the dynamic and multivariate behavior of the plant away from its steady-state operation. The classical analysis techniques, such as response function fits,^{357,358} or chemometrics approaches³⁵⁹ give some insights into the typical response to the different perturbations. However, these techniques cannot take the full multivariate, non-linear nature of the time-dependent behavior of a complex plant into account. In addition, conventional causal analysis techniques cannot be used without an understanding of the mechanisms (i.e., the causal graph)³⁶⁰ or additional experiments, which interpretation, however, is not trivial since one cannot easily compare to a baseline (i.e., the behavior of the plant under the same environmental and solvent conditions without a particular change).

In this work, we show that data science methods that are typically used for dynamic pattern recognition and predictions of financial data can successfully be adapted to forecast the performance of a plant (in real-time) given its current and historic behavior, even if it is operated far from its steady-state conditions—without a detailed understanding of the underlying process. These forecasts can subsequently be used to model potential emission mitigation scenarios and to understand experimental observations.

5.1.1 Experimental campaign

To mimic the intermittency expected for the operation of future power plants, we carried out an experimental campaign that involved a series of stress tests on the pilot capture plant at Niederaußem. Importantly, due to its size and the fact that it has been operating on a slipstream of flue gas from a raw lignite-fired power plant^{361,362} with CESAR1 solvent for over 12 months (see Figure 29 for a schematic flow diagram), it provides an ideal real-life example of the difficulties of understanding amine emissions.³⁵⁸ These stress tests were based on eight different scenarios of how intermittency can impact the operation and hence the amine emission of the capture plant (see Appendix E.1 for more details on the rationale of these scenarios).

Figure 30 shows that the sequence of stress tests causes emissions significantly higher than those under normal operating conditions. Another interesting observation is that piperazine (Pz) and 2-amino-2-methyl-1-propanol (AMP) have different emission profiles. Such a campaign gives us a wealth of experimental data on the behavior of a capture plant. Clearly, these data would be more valuable if we could use

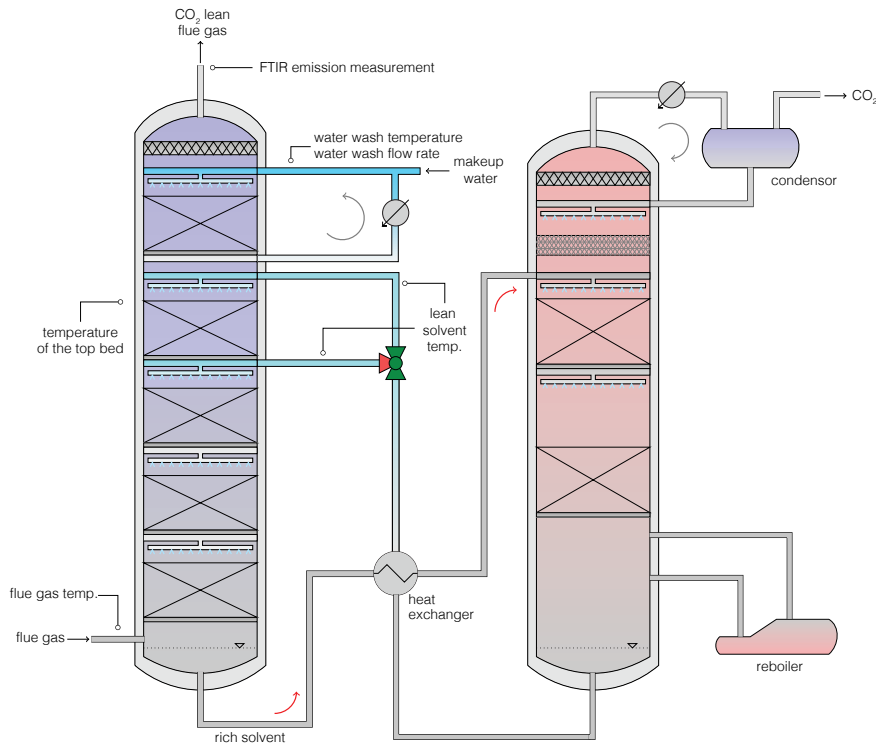


Figure 29: Simplified process flow diagram of the post-combustion carbon capture pilot plant at Niederaußem. The plant uses a slipstream from the coal-fired power plant. The positions of the process parameters discussed in the main text are indicated in the figure. A complete piping and instrumentation diagram (P&ID) can be found in Figure 147.

them for quantitative predictions on future emissions. However, we cannot even make qualitative predictions. For example, during most of the stress tests, interventions of the operators were required to ensure the safe operation of the plant. Such interventions make even a qualitative interpretation of the data a challenge, as we cannot disentangle the effects of these interventions from the operational changes induced by the stress test.

Therefore, we have a case in which we have a large amount of valuable experimental data, but where the complexity of the operation of the pilot plant does not allow any other conclusion that these emissions are problematic. In particular, we cannot draw any statistically relevant conclusions on why our stress tests caused such a dramatic increase in emissions and which countermeasures we could take to reduce emissions.

5.1.2 Machine learning model

During the experimental campaign, data were taken every minute. This provides us with a large data set. Such a dataset allows us to use data-science methods and develop a machine-learning model to analyze the data. In this section, we summarize the main features of our approach; for details, we refer to the Appendix and the methods section. Our machine-learning approach is based on the observation that we can build a forecasting model by thinking of the time-dependent process and emission data as an image (i.e., matrix of data, see Figure 31). This representation allows us to use the most powerful machine-learning techniques for pattern recogni-

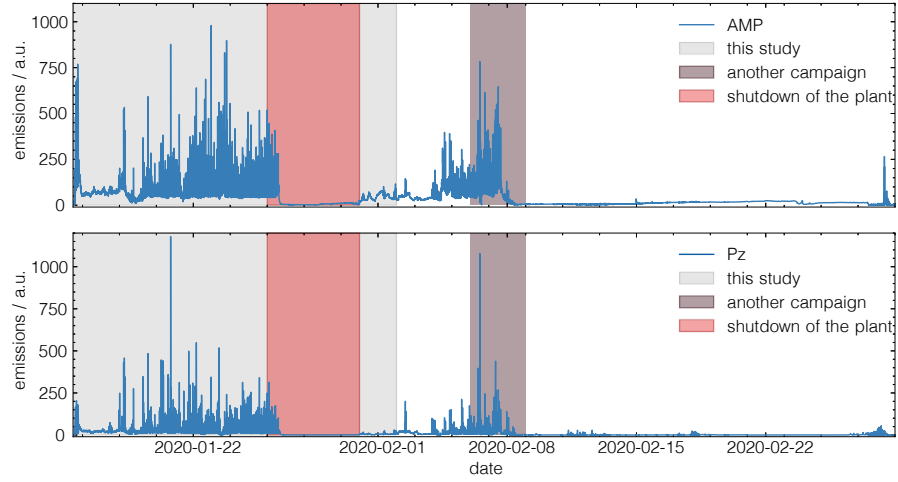


Figure 30: Amine emissions during and after the experimental campaign. The time frame of the stress tests is highlighted in grey. The power plant was shut down from 25 to 30 January (red region), which explains the very low emissions around that time. In this work, we only used the data generated before the shutdown of the plant. In the period of 6–8 February (grey region) other experiments were carried out at the pilot plant, but these were not part of our campaign. This figure shows that applying the different scenarios causes the plant to emit much more compared to its steady-state operation. A preliminary analysis of the data has been reported in Charalambous et al.³⁵⁸

tion. In this representation, the state of the plant at a given time t defines a “state” feature vector $\mathbf{x}(t)$ with p elements representing the process variables (e.g., flue gas temperature, water wash temperature). If we take the state vectors of the plant for t timestamps, we have a matrix of $t \times p$ entries, which can be seen as an “image” that is connected to a future emission profile, $\mathbf{y}(t)$.

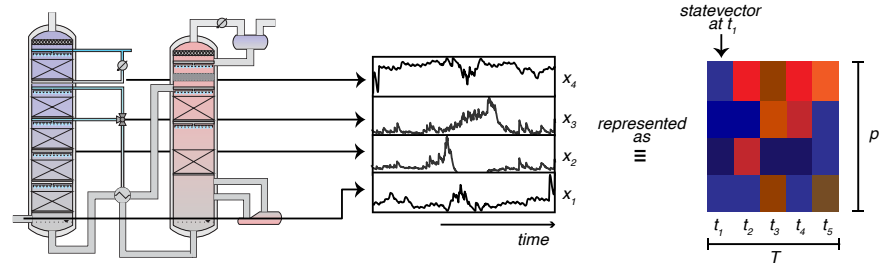


Figure 31: Schematic illustration of the data representation. The data set can be thought of an “image” with “width” equal to length of the input sequence (T) and “height” equal to the number of parameters, p . We represent the value of parameter x_j at a time t_i with colors. We then use a machine learning model to learn how this “image” characterizing the history, and current state of the plant is connected to its future emissions.

The next step is to link the pattern in the image of the history of the plant to a particular future emission. For this, we have adopted a gradient-boosted decision tree^{363,364} model that is trained on a feature vector of concatenated historic data of process parameters and emissions (i.e., we combine the rows, characterizing the different parameters and emissions, into a long vector). We train these models using quantile loss^{365,366} to obtain uncertainty estimates. We have also adopted a temporal CNN with Monte Carlo Dropout for uncertainty estimation and show

results (equivalent to those obtained with the gradient-boosted decision tree) obtained with this model in Appendix E.8.

5.1.3 Insights into amine emissions from machine learning

We apply our machine learning model for different purposes, and each of them requires us to forecast the emissions, but each with a different aim and time horizon:

1. *(Real-time) prediction of future emissions:* The aim here is to predict what the emissions x hours in the future will be given the historic and current operation and emissions.
2. *Causal impact analysis of the data:* To measure the impact of a particular stress test on the amine emissions, a reference is needed; i.e., a baseline that gives us the emissions that would occur without the changes directly induced by the stress test. Without this baseline, it is impossible to correctly quantify the effect of the different stress tests on the observed emissions.
3. *Emissions mitigation:* To understand and identify how we can mitigate emissions, we use our model to predict emissions in “what-if” scenarios. For example, we predict how the overall emissions would change if we ran the entire experimental campaign with a lower temperature of the water wash section.

In the next sections, we show how we use our machine learning model to forecast amine emissions for the three different purposes mentioned above. The basic model architecture we use is the same; however, the way we apply and train the model for the different ways of forecasting is different.

5.1.4 Prediction of future emissions

The machine learning model we introduced in the previous section uses historic data to predict future emissions. For example, we use a sequence of input data (e.g., 2 hours), and predict the emissions, say, 10 minutes, 1 hour, or 2 hours in the future. For doing so, we use a sliding window; for the next prediction, we update the input sequence with the observed emissions (see Figure 32). The model can be used for making predictions for any time horizon; however, one can expect the accuracy to decrease for longer time horizons compared to shorter ones. To quantify the accuracy of our prediction, we use the data we have not used in our training (and validation) set. One has to be careful in making this comparison. Our machine learning model makes predictions on the likely emissions, given the plant data preceding these predictions, and in the testing step, we use the measured data in the test set. However, this validation is overly pessimistic with respect to potential real-world applications since the validation and test set contain, by construction, step changes that have not been seen in the training set. In addition, these stress tests are designed to take the plant outside normal operations. Also, the moment such a stress test is applied has no logical relation with the historic data of the plant and hence cannot be learned. This makes our validation overly pessimistic, as we are rather testing how well our learning extends to very extreme conditions in the stress test.

5.1.5 Causal impact analysis

The key motivation for performing our experimental campaign is to understand what changes to the plant’s operation have a significant impact on the amine emissions. This understanding is essential to identify those parameters that must be tightly

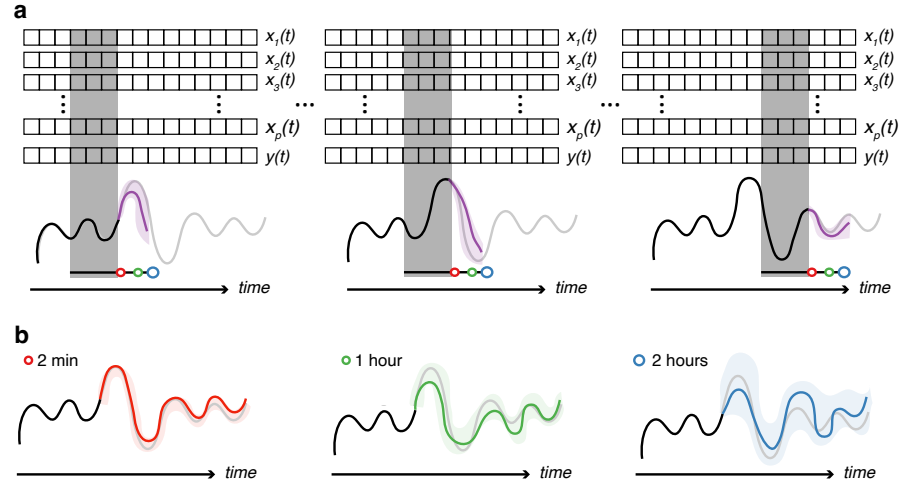


Figure 32: Predicting future emissions. In this figure, $x_i(t)$ represents the input data of the plant (e.g., temperatures, pressures, etc. in different parts of the plant) and $y(t)$ the emissions. The grey box represents the data the model uses to predict future emissions. The black curve represents the measured past emissions, and the grey curve the measured future emissions. The purple curve represents the “real-time” predicted future emissions, and the shaded purple area is the uncertainty of the predictions. We mimic such “real-time” predictions by sliding our grey box over the data; i.e., the measurements of the current time are added, and the oldest data are no longer seen by the window, and we make a new prediction. In the bottom figure, we collect the predictions for the different time horizons (2 minutes red, 1 hour green, and 2 hours blue). We use the first half of the dataset to train the model.

monitored and controlled to mitigate emissions. In statistics, the gold standard for answering such a question requires a control experiment³⁶⁷ to establish a baseline. At present, such a baseline is impossible to obtain. As the pilot plant receives the flue gas from a commercially operated coal-fired power plant, it is impossible to precisely reproduce the varying conditions of the plant. For this, one would need to run two identical pilot plants simultaneously.

Similar problems exist, for example, in finance, where one might want to measure the impact of some political intervention and where it is equally impossible to duplicate society for a control experiment. Interestingly, for these problems, causal impact analysis³⁶⁸ can be used to construct a so-called counterfactual baseline of the system’s behavior *without* the intervention. For this, we use our machine-learning model to “rerun” the campaign, but without the stress tests. The fact that we now can obtain a reasonable performance baseline is one of the major technical insights of our approach.

Let us assume that we have a perturbation on variable $x_2(t)$, e.g., we apply a step change for variable $x_2(t)$ at times $t \in [t_{\text{startstep}}; t_{\text{endstep}}]$. To obtain a prediction of the baseline, we then train our model on the training data but *without* any input from $x_2(t)$ [i.e., we also remove (Granger) causally related features]. We then have a model that predicts the emissions worse than if $x_2(t)$ was included in the training (as fewer features are used as input for the model), but it does give us our best prediction for the normal operation of the plant *irrespective* of the actual value of $x_2(t)$. This is the best approximation of the baseline operation we are interested in. Similarly, we train our model for all other variables that are changed during the different stress tests conducted in the experimental campaign. We then use each of these models to predict the baseline (see Figure 33).

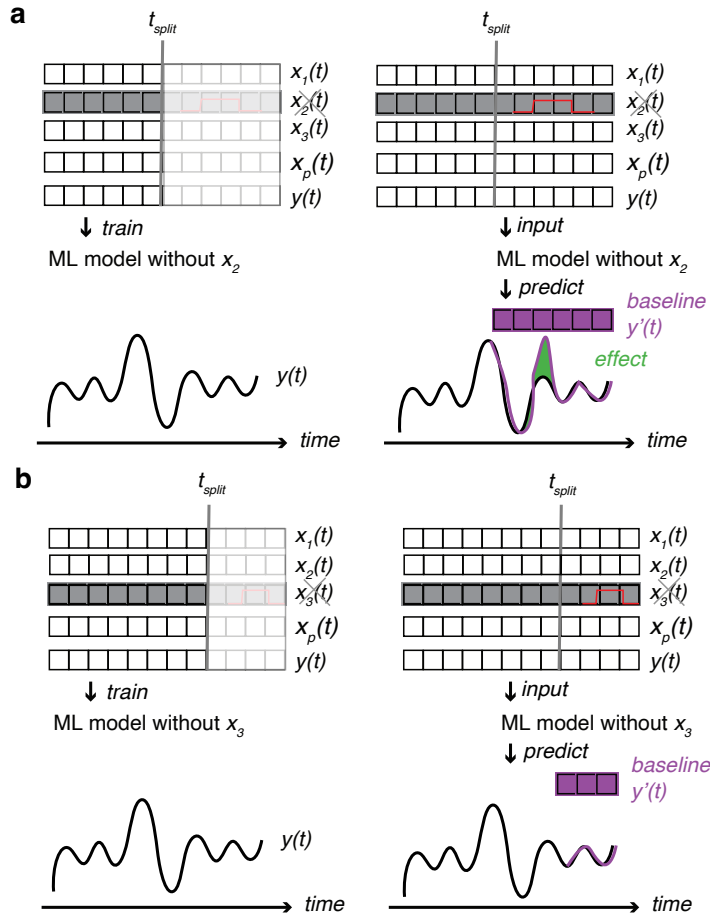


Figure 33: Causal impact analysis. The left column shows the training of the model, in which we use all the data preceding a particular step change to train our model on predicting the capture plant's performance. In this example, we make a step change of variable $x_2(t)$, for example, **a**, and we train a model without variable $x_2(t)$. For example, **b**, we make a step change in variable $x_3(t)$ and hence train another model without variable $x_3(t)$. In this calculation, we have assumed that the other variables are not causally related with $x_2(t)$ and $x_3(t)$, respectively; if there is a causal relation, these variables also need to be removed. The right column shows how we compute the baseline; the step changes are indicated by the vertical lines. The black curve gives the actual plant data, $y(t)$, obtained from the experimental campaign. The violet curves, $y'(t)$ give the machine-learning predictions of normal operation without the stress test, i.e., the baseline. The predictions show that the x_2 step change test caused a real reduction in emissions, whereas the change in x_3 showed no effect.

5.1.6 Emission mitigation

To shed light on how we can reduce the overall amine emissions during, for instance, a given experimental campaign, we have used our model to run “what-if” scenarios. These scenarios were inspired by the outcome of the causal impact analysis, which highlighted some of the variables that impacted the emissions the most. An example of such a scenario could be: “what-if we run the entire stress test with an increase in variable $x_2(t)$ of 10 %”. For this scenario, we replace the input of our model $x_2(t) \rightarrow 1.1x_2(t)$ (see Figure 34) to compute the predicted emissions $y'(t)$. We can then

compute the change in total emissions from:

$$\text{rel. emission change} = 100 \times \frac{\int dt (y'(t) - y(t))}{\int dt y(t)}.$$

To compute these scenarios, we need to predict the emissions $y(t)$ given an input $x(t)$. To do so, we retrain the model using all available data from the experimental campaign to ensure the highest possible accuracy from our model. For a more detailed discussion, see also Appendix E.11.

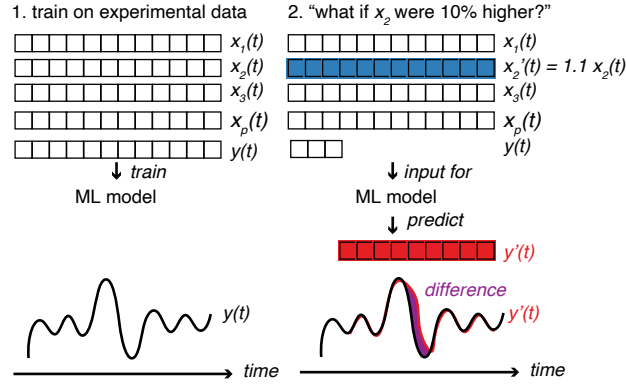


Figure 34: Emission mitigation. To predict the effects of a given variable on the total emissions, we train our machine-learning model on the entire data set (left). We then use this model to run a “what-if” scenario, i.e., to use the model trained in the first step to predict the emissions $y'(t)$ (red) for this changed input. For example, what are the emissions if we replace input $x_2(t)$ (blue array) by, say, $x'_2(t) = \alpha_2(t)$ (here, $\alpha_2 = 1.1$)? We can then calculate the difference between the actual measured emissions $y(t)$ and the predictions $y'(t)$ (green). If we perform this for different α , we can estimate and plot the change in emissions as a function of α .

5.2 RESULTS AND DISCUSSION

5.2.1 Prediction of future emissions

In Figure 35, we compare the measured Pz and AMP emissions with the predicted emissions for different forecasting horizons. For the short-horizon predictions (top row), we observe that the measured emissions are typically within our prediction interval (shaded area) and that our model even correctly captures the spikes in the emission profile [AMP mean absolute percentage error (MAPE) 2.4 %, overall percentage error (OPE) 0.38 %; Pz MAPE 4.3 %, OPE 2.0 %, see Appendix E.7]. We can also make predictions for a longer time horizon. For one to two-hour windows, we can correctly forecast the trends, but as expected, we lose accuracy on the events (such as spikes) that happen on a short timescale [AMP MAPE 9.5 %, OPE 3.8 %; Pz MAPE 21 % OPE 10 %]. It is interesting to zoom in on some areas where our predictions deviate significantly from the actual measurements. These deviations are associated with a stress test not seen in the training, yet it is encouraging to see that our model did learn something as we predict the trends. In addition, our model indicates at those conditions a substantial uncertainty, which is exactly how the stress test is designed; to take the plan far outside normal operations. Therefore, it is very encouraging that our model recognizes this and correctly reflects this in the uncertainties.

Of course, a stress test is far from ideal to test our model to make (real-time) predictions, but these results do indicate that our model, if applied under normal operating conditions, can be used to make predictions about the emissions on a two-hour window, which does give the operators a window to take actions if emissions are predicted to exceed specification limits.

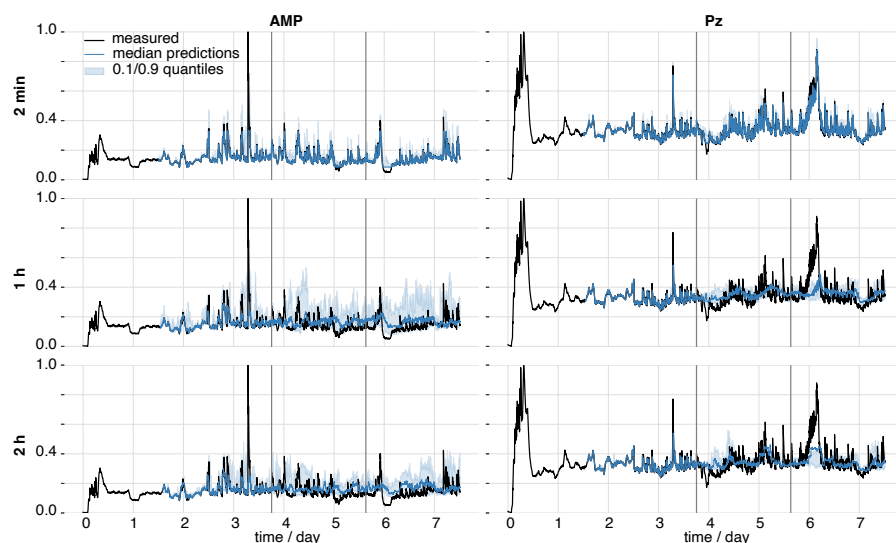


Figure 35: Amine emissions as predicted by the machine learning model. To test the performance of the model for the amine emissions of AMP (left) and Pz (right), we trained the model on the first part of the data, used a subsequent part for hyperparameter search, and tested the performance on the final part. The splits are indicated with grey vertical lines. The gap without predictions is due to the fact that the model needs to be initialized with a part of the sequence. The blue lines show so-called historical forecasts, which can be produced by an expanding window approach where the model is moved over the time series, and we simulate what the predictions would have been if one used the model with the forecasting horizon with an updated dataset (i.e., the model sees the actual emissions for making forecasts and does not have to use its predictions, but we do not retrain the model). In the rows, we show the predictions for different forecasting horizons, and we can observe, as one would expect, that the predictions for shorter forecasting horizons are better than those for longer ones. The shaded areas fill the range between the 10 % and 90 % quantiles.

5.2.2 Causal impact analysis

The first step in our analysis of the experimental data is computing the baseline for all stress tests. In Figure 36, we compare the measured emissions for three of the stress tests with the predicted baseline (which we predicted with the model architecture we validated in Figure 35). We can see the importance of these baselines in Figure 36 **c** and **d**. At the first black vertical line, the lean solvent temperature was increased from 43 °C to 52 °C and put back to normal at the second vertical line. The measurements (black lines) suggest increased emissions during and after the intervention. However, this behavior is strikingly similar to the prediction without the intervention (within the prediction interval) for AMP. Interestingly, applying the same analysis to those stress tests that involved changes in the water wash flow rate or the solvent and water wash temperature (**a**, **b**, **e** and **f**), we observe a significant effect. In Appendix E.10, we show the analysis for all interventions investigated in our campaign.

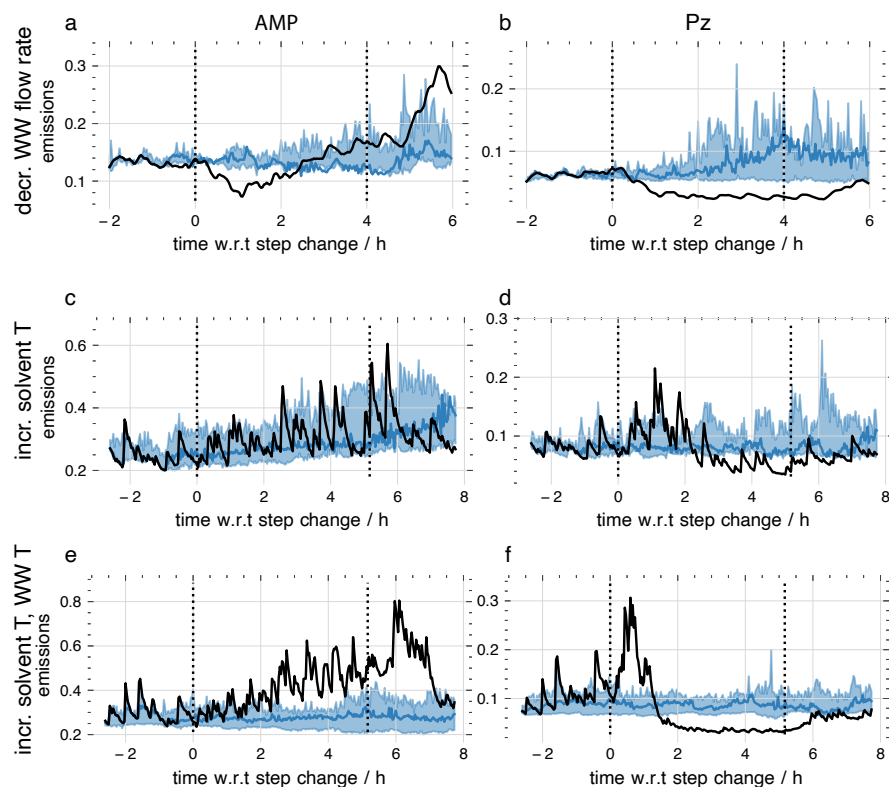


Figure 36: Causal impact analysis for three of our dynamic experiments. In a causal impact analysis, we use the machine learning model to predict what the emissions (**a**, **c**, **e** AMP emissions, and **b**, **d**, **f** Pz) were without intervention (blue). The effect size is the difference between the prediction and the actual measurement (black). If we observe no difference between the measurement (black) and prediction (blue), then there is no effect. **a**, **b** Shows the measurement and predictions for the step decrease (decr.) in water wash (WW) flow rate. One can observe that also the counterfactual model forecasts an increase (incr.) in amine emissions compared to the actual observations. **c**, **d** Shows the effect of the increase (incr.) in lean solvent temperature. One can observe that Pz (**d**), in contrast to AMP (**c**), shows a significant reduction in emission w.r.t. the baseline. **e**, **f** Shows the effect of increased water wash and lean solvent temperature. One can observe that Pz (**f**), in contrast to AMP (**e**), shows a reduction in emission w.r.t. the baseline. Shaded areas cover the area between the 0.05 and 0.95 percentile. Dotted vertical lines indicate the start and end of the step change.

Interestingly, the causal impact analysis reduces this extremely complex emission behavior (see Figure 30) into a surprisingly simple conclusion that controlling the water wash and solvent temperature and the water wash flow rate are the most promising handles for emission mitigation. However, without the counterfactual baseline, we would have concluded that many other interventions that show a change in emissions during the intervention are also good handles for emission control. This shows how machine-learning techniques can be used to extract insights from complex experimental datasets that remained opaque to conventional approaches.

5.2.3 Emission mitigation

The causal impact analysis can give us insights into the significance and magnitude of the effects of changes we actually performed on the plant. However, many other parameters were implicitly changed during the stress test. Using our model, we can use this data to investigate which changes to the operation of the plant would result in lower overall emissions during the stress test.

Figure 37 shows the predicted cumulative change in amine emissions over the full campaign for the two sets of variables that caused some of the largest changes in our *in silico* experiments. In these *in silico* experiments, we change the value of two parameters by a fixed percentage over the entire stress test, keeping the dynamics unchanged, and let our model predict the emissions. The heatmaps then show the difference with the measured emissions, for which reason the center (0,0) of the heatmaps is grey. These figures point to the most important conclusion from our experimental campaign. Figure 37a suggests that lower AMP emissions are obtained when operating at a lower solvent temperature. However, we do not have the minimum Pz emissions under these conditions. On the other hand, minimum Pz emissions are predicted for increased lean solvent temperature and increased temperature at the top bed—under which conditions AMP emissions are predicted to increase. Similar conclusions can be drawn from the other scenarios (see Appendix E.11). These results suggest that Pz and AMP have different emission mechanisms. If volatility were the only mechanism, one would expect the amine emissions to increase with increasing solvent temperatures. This is what we observe for AMP. Since AMP is more volatile, the AMP partial pressure throughout the column is expected to be around two orders of magnitude higher than that of Pz³⁶⁹. One would not expect significant emissions of Pz if volatility were the only mechanism. However, one can also have emissions through aerosols³⁷⁰. These aerosol emissions are thought to be related to supersaturation in the column, which can be caused by a temperature bulge in the column profile that can be influenced by a change in lean solvent temperature^{351,371}. Absorbed in such aerosol droplets, Pz and AMP are forming non-volatile carbamates, and (pure component) studies have shown that the kinetics of this reaction is much faster for Pz.³⁶⁹ Moreover, due to steric hindrance, the AMP carbamates are short-lived, and AMP is present as a protonated species in equilibrium with the free amine.³⁷² This leads to a situation in which there is a back-pressure build-up that hinders further AMP absorption in aerosol particles, which is not the case for Pz. Hence, one would expect the aerosol mechanism to be more relevant for Pz emissions, and we conclude that in our stress test, the aerosol mechanism seems more relevant for Pz than for AMP. Because the two components in the CESAR1 mixture have different governing emission mechanisms, different mitigation strategies have opposite effects on the emission of the two components. Therefore, one needs to design the capture plant to be able to deal with both mechanisms. This is a more challenging task when considering blended solvents, such as CESAR1, than single amine solvents. Including these additional costs in the current discussion is essential to replace conventional MEA-based capture plants with those based on more advanced solvent systems such as CESAR1. Even though we could not have derived this insight without our machine-learning-based analysis, further experiments will be needed for a more detailed understanding of the causal mechanisms more experiments as our current model can only highlight predictive correlations.

Even at steady-state, we would not have been able to develop a conventional process model to predict amine emissions from the carbon capture plant. For instance, we would need additional experiments as we lack relevant thermodynamic data on the amines and an understanding of the emission mechanisms. To make things worse, the plant was far from steady-state over the course of the experimen-

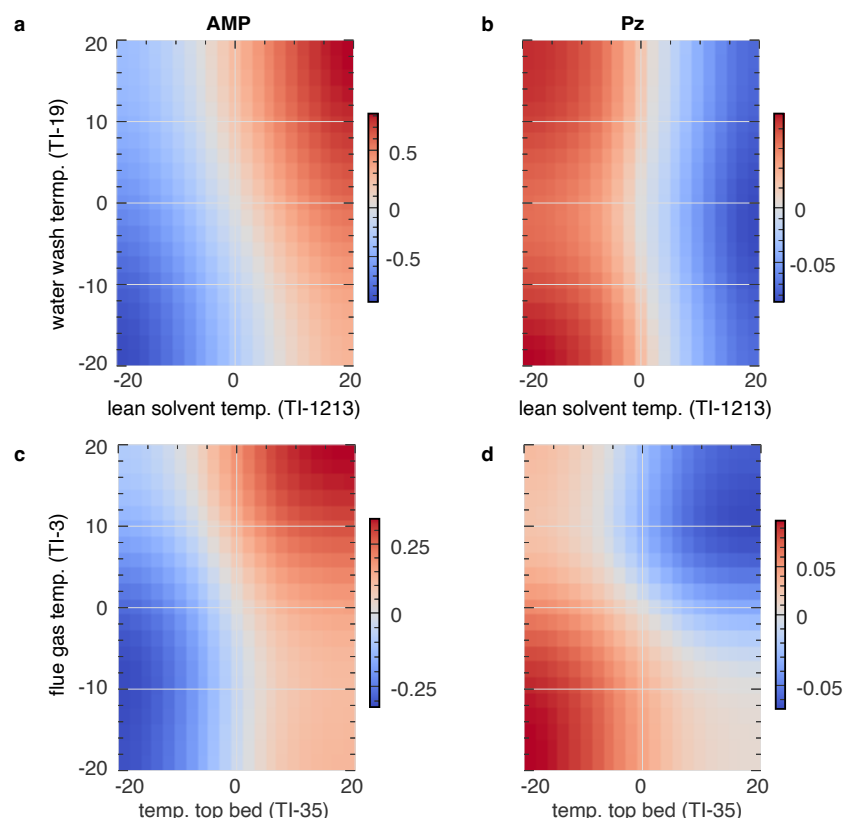


Figure 37: Predicted changes in emissions. Ordinate and abscissa show the relative change in the process variable (in percent). The color indicates the cumulative change in normalized emissions over the full observation time compared to the actual emissions (i.e., not absolute emissions). To increase the reliability of the forecasts, we trained the model for this analysis on the complete dataset and used a short output sequence length. Left column (a, c) shows predicted changes in emissions for AMP. The right column (b, d) shows the predicted changes in emissions for piperazine (Pz). a, b plot the changes in emissions for changes in water wash temperature (temp.) and lean solvent temperature. For AMP, the highest predicted reduction in emissions is for decreased water wash temperature and decreased temperature of the lean solvent, whereas for Pz the highest predicted reductions in emissions are possible for increased solvent temperature. c, d shows the predicted change in emissions for a change in the temperature of the flue gas upstream of the adsorber column and the temperature of the top bed. Here, decreasing the temperature at the top bed and the temperature of the flue gas yields the lowest AMP emissions but high predicted Pz emissions.

tal campaign. The current process models are too simple to deal with this complexity. In this work, we developed an alternative approach in which we start with the data and learn the mapping between the process and the emissions directly from the data. The resulting machine-learning model allows us to not only forecast (in real-time) the emissions of the plant but also to gain insights into which parameters are key for emission mitigation. A similar approach can be used to forecast and understand other key performance parameters, such as those related to the plant energy requirements. Amine emissions from a carbon capture plant are just one example of an industrial process for which a better understanding of its operation beyond its steady state is needed. Another example is the start-up of a plant during which one has to carry out many tests to identify safe operational limits. These tests can take many months before a plant can be put into operation. Typically, during such

a start-up phase or any other change to a new operating regime, there is a lot of data created and collected, but this data collection has outpaced our ability to sensibly analyze the data, let alone understand it. Our work shows that we could feed the data into an active learning model to harvest all the knowledge collected during these experiments. Interrogation of this model can help us define the next most informative experiment,^{207,373} which we expect to greatly reduce the time to operability and, in contrast to conventional approaches, can easily (via retraining) adjust to changes in the plant (e.g., solvent degradation). This power of machine learning in chemical engineering also highlights the need to share data in a machine-actionable form,^{38,374,375} and we believe that machine learning can potentially make an even bigger impact in chemical and process engineering than it did in computer vision. In the case of computer vision, the basic features of an image learned by a model are often closely related to how we perceive images with our brains. However, in an industrial plant, we often lack an understanding of the underlying mechanisms, but with machine learning, we can discover the underlying rules of the mapping from the parameters to observables and make predictions for phenomena we could not predict thus far.

5.3 METHODS

5.3.1 Pilot plant

Figure 29 shows a schematic flow diagram of the capture plant at Niederaußem (Germany). The flue gas is supplied by a 965 MW_{el} raw lignite-fired power plant subjected to a state-of-the-art multistage electrostatic precipitator, a conventional wet limestone flue gas desulphurization plant, and a direct contact cooler (DCC) located upstream of the absorber. The capture plant follows a conventional amine scrubbing process. The absorber column comprises four beds and is integrated with a flexible intercooling system and a water wash section. The flexible intercooler, which can be located either between the bottom and the second packing or between the second and third packings, controls the temperature rise in the absorber. A water wash section has been added to the pilot plant to reduce amine emissions to the atmosphere.^{376,377} The amine degradation, due to the presence of oxygen and other impurities such as nitrogen oxides, as well as elevated temperatures during solvent regeneration, can result in other gaseous emissions of degradation compounds such as ammonia.³⁷⁸

The flue gas upstream of the absorber was analyzed using a BA5000 Bühler infrared spectroscope. The CO₂-lean flue gas downstream of the water wash outlet was analyzed using a GasMET CX/DX 4000 analyzer (i.e., CO₂, CO, O₂, AMP, Pz, NH₃, and H₂O).

5.3.2 Experimental campaign: Intermittency scenarios

As the baseline, we assume that the capture plant operates with the power plant at full load but that the intermittency associated with a future increase in renewables will cause regular changes in the load of the power plant. Variations in this load not only change the amount of flue gas that the capture plant has to process but can also change the amount of steam available for the capture plant. In the scenarios that drive our stress tests, we focus on those (combinations of) changes, of which our previous study on MEA³⁶¹ has shown that they can impact emissions. The timescale and the magnitude of the changes are based on the expected intermittency³⁵³ and

typical requirements of the grid services,^{356,379,380} respectively. A more detailed description is given in Appendix E.1.

5.3.3 Machine learning

To avoid overfitting and the exploitation of spurious correlations, the models were trained on a small feature set created using manual feature selection and engineering (see Appendix E.6). For all our modeling, we removed deterministic trend components from the data using linear regression, which is motivated by the fact that the characteristic timescale of these components is beyond the one captured by our dataset (and analysis). Additionally, removed outliers using a z-score filter ($z = 3$), performed exponential window smoothing (window size 16 min), and downsampled the data to a frequency of 2 min. The impact of the preprocessing is shown in Figure 150. For use in the models, we additionally standardized the data using min-max scaling. We did not retrain models for historical forecasts.

Quantile Regression using Gradient Boosted Decision Tree Models

To forecast the emissions, we used gradient-boosted decision tree models in which the feature vector is constructed by concatenating lagged time series for process parameters and emissions. In this approach, we train a new gradient-boosted decision tree (as implemented in the LightGBM library³⁸¹) for every forecasting horizon using the darts package³⁸². To obtain uncertainty estimates, we use quantile regression. We tune the hyperparameters of the gradient-boosted decision tree and the number of lags using hyperparameter optimization on a validation set using Bayesian optimization. For all models, we scaled the data (emissions and process variables) based on statistics computed on the training dataset.

Causal impact analysis

For the causal impact analysis, we remove causally related covariates and trained models on the data of the days preceding the step change and following the step changes. For every model, we performed a new hyperparameter optimization (using the shorter sequence preceding or following the step change as a validation set). We also attempted to use Bayesian structured time series models as in the original implementation of the causal impact analysis techniques³⁶⁸ and found qualitative agreement.

We made use of the following Python³⁸³ libraries: pandas,²⁸⁹ sklearn,²⁷⁹ scipy,²⁹⁰ statsmodels,³⁸⁴ matplotlib,²⁸⁴ jupyter,²⁸² numpy,³⁸⁵ pytorch,³⁸⁶ darts,³⁸² lightgbm,³⁸¹ shap.³⁸⁷

DATA AND CODE AVAILABILITY

The raw data (emissions and process parameters) and model checkpoints needed to evaluate the conclusions in this paper are archived on Zenodo (DOI: 10.5281/zenodo.5153417). The code for our analysis is available on GitHub (github.com/kjappelbaum/aem1) and archived on Zenodo (DOI: 10.5281/zenodo.7116093).

Part III

DISCUSSION AND OUTLOOK

6

IS GPT-3 ALL YOU NEED FOR
LOW-DATA DISCOVERY IN
CHEMISTRY?

ABSTRACT Machine learning has revolutionized many fields and has recently found applications in chemistry and materials science. The small datasets commonly found in chemistry lead to various sophisticated machine-learning approaches that incorporate chemical knowledge for each application and therefore require a lot of expertise to develop. Here, we show that large language models trained on vast amounts of text extracted from the internet can easily be adapted to solve various tasks in chemistry and materials science by simply prompting them with chemical questions in natural language. We compared this approach with dedicated machine-learning models for many applications spanning properties of molecules and materials to the yield of chemical reactions. Surprisingly, we find this approach performs comparable to or even outperforms the conventional techniques—in particular in the low data limit. In addition, by simply inverting the questions, we can even perform inverse design successfully. The high performance, especially for small data sets, combined with the ease of use, can have a fundamental impact on how we leverage machine learning in the chemical and material sciences. Next to a literature search, querying a foundation model might become a routine way to bootstrap a project by leveraging

the collective knowledge encoded in these foundation models.

CITATION Prior versions of this chapter have been presented at the 2022 NeurIPS AI4Mat and “Critical assessment of molecular machine learning” workshops. A current version is available as a preprint on ChemRxiv: Jablonka, K. M. et al. In *ChemRxiv preprint 10.26434/chemrxiv-2023-fw8n4*, 2023.

CONTRIBUTION K.M.J performed and designed the experiments and wrote the article with editing by and contributions from B.S. and feedback from P.S. A.O. provided support with the DFT simulations.

6.1 INTRODUCTION

One of the fascinating advances in machine learning has been the development of extremely large language models (LLMs), so-called foundation models.^{389–393} These models are appealing because of their simplicity; given any text prompt, like a phrase or a sentence, these models return text that completes the phrase in natural language. Interestingly, the quality of the return text is so high that, in many instances, one cannot even tell that a machine wrote it. We only start to see the impact of this as many (startup) companies are focused on creating apps for a particular application. From a scientific point of view, the most striking examples are that these foundation models can write sensible abstracts for scientific articles or even code for particular programming tasks.^{394–397} Recently, it has been shown that these models can also solve relatively simple tabular regression and classification tasks.³⁹⁸ But as these models were not explicitly trained on these tasks, it is a remarkable result.³⁹³

That these models can solve simple tasks, they are not trained for made us wonder whether they can also answer scientific questions for which we do not have an answer. As most chemistry problems can be represented in text form, we should be able to train these models to answer questions that chemists or material scientists have. For example, “if I change the metal in my metal-organic framework, will it be stable in water?” Or, “what is the band gap of my material?” These questions are often impossible to answer using theory or require highly sophisticated simulations or experiments.

We will always have very little (experimental) data for applications in chemistry and material science. Hence, it is important that this learning does not require millions of data points but that meaningful results can already be obtained with tens to hundreds of data points. We know from prior work on text classification or generation applications that this works particularly well using models from the GPT-3 family,³⁹³ which were trained by the artificial intelligence company OpenAI. The largest GPT-3 model has approx. 175 billion parameters were trained on hundreds of billions of text fragments (tokens). In this work, we show that this model gives a surprisingly good performance for a range of very different chemistry questions (Figure 38), often outperforming the state-of-the-art machine learning models specifically developed for these tasks.

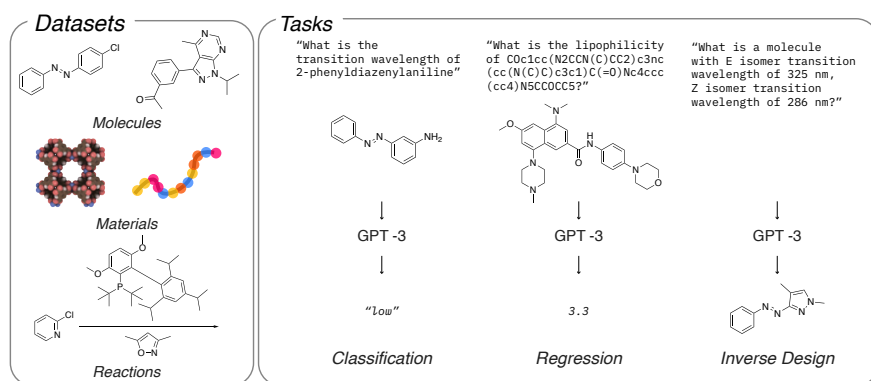


Figure 38: Overview of the datasets and tasks addressed in this work. In this work, we benchmark GPT-3 on datasets spanning chemical space from molecules over materials to reactions (see Appendix F.2). On those datasets, we investigate different tasks ranging from classification, i.e., predicting a class (e.g., “high”, “low”) given a text representation of a molecule, material, or reaction, regression, i.e., prediction of floating point numbers, to inverse design—the prediction of molecules. MOF rendering created with iRASP. ³⁴⁶

We selected a set of questions that illustrate that these models can answer a wide range of scientific questions ranging from the properties of materials, how to synthesize materials, and even how to design materials. In selecting these questions, we included some that have been addressed with machine learning. This allowed us to benchmark against state-of-the-art machine learning approaches specifically developed for these applications.

Before discussing the different applications in detail, let us first discuss how we fine-tune³⁹⁹ the GPT-3 model in practice. For this, let us look at a simple but highly non-trivial example. High entropy alloys have attracted much interest as a novel class of structural metals. Interestingly, one has a sheer infinite number of possible combinations of metals. From a practical point of view, it is important to know if a given combination of metals will form a solid solution or multiple phases. Hence, the question we would like to ask is "What is the phase of <composition of the high entropy alloy>?" and our model should give a text completion from the set of possible answers {single phase, multi-phase}. In Table 3, we have given the set of questions and answers we have used to fine-tune the GPT-3 model. These are questions and answers on high entropy alloys for which the phase has been experimentally determined. The model tuning takes a few minutes and gives us a new model, which takes as input "What is the phase of Tb0.5Y0.5" and gives as text completion "1", which corresponds to single-phase. This simple example already gives some remarkable results. We selected this example to directly compare its performance with the current state-of-the-art machine learning models with descriptors specially developed to mimic the relevant chemistry for this application.⁴⁰⁰ In Figure 39, we show that with only around 50 data points, we get a similar performance as the model of Pei et al.⁴⁰⁰, which was trained on more than 1000 data points.

Table 3: Example prompts and completions for predicting the phase of high-entropy alloys.

These models have been trained using a self-supervised approach, i.e., to predict the next token given an input text sequence. This implies we offer the list of questions and answers as one large string. The program learns that in our string "###" indicates the end of a prompt and "@@" the end of a completion. Here, we used the fact that it is cheaper and easier to learn one character, hence 0 = multi-phase. If this training string is submitted to the GPT-3 API, one gets the identifier for the fine-tuned model. With this identifier, one can query the GPT-3 API for the completion of an unknown high-entropy alloy.

prompt	completion	experimental
What is the phase of Co1Cu1Fe1Ni1V1?###	0	multi-phase
What is the phase of Pu0.75Zr0.25?###	1	single-phase
What is the phase of BeFe?###	0	multi-phase
What is the phase of LiTa?###	0	multi-phase
What is the phase of Nb0.5Ta0.5?###	1	single-phase
What is the phase of Al0.1W0.9?###	1	single-phase
What is the phase of Cr0.5Fe0.5?###	1	single-phase
What is the phase of Al1Co1Cr1Cu1Fe1Ni1Ti1?###	0	multi-phase
What is the phase of Cu0.5Mn0.5?###	1	single-phase
What is the phase of OsU?###	0	multi-phase

These results made us wonder if similar results can be obtained for other properties. Hence, we looked at a range of very different properties of molecules, materials as well as chemical reactions, i.e., spanning most, if not all, aspects of chemistry. We focused on those applications for which conventional machine-learning methods have been specifically developed and generally accepted as benchmarks in their field. In addition, we also compared our model to the top-performing ones on tasks from the Matbench²¹² suite of benchmarks. Matbench has been specifically created

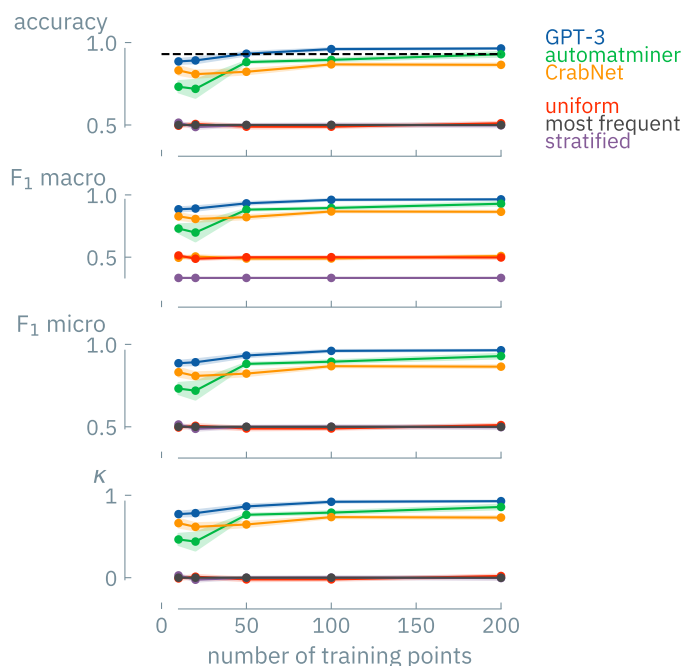


Figure 39: Performance of our GPT-3 model for predicting solid-solution formation in high-entropy alloys. In the top figure, we compare the accuracy of the model as a function of the number of training points. The dashed horizontal line indicates the performance reported in Pei et al.⁴⁰⁰ using a dataset of 1252 points and 10-fold cross-validation, i.e., corresponding to a training set size of around 1126 points. The orange (uniform), purple (most frequent), and red (stratified) are baselines that one would obtain without any learning. The green line is the results we obtained using the Automatminer,²¹² which uses as input the chemical composition. The Automatminer then returns the best featurization and model among those implemented using automated machine learning with genetic programming (as implemented in the TPOT package²¹⁵). We additionally tested a neural network, compositionally-restricted attention-based network (CrabNet) (yellow line),⁴⁰¹ that performs well using compositions as input. The blue line is the performance of our GPT-3 model (with error bands showing the standard error of the mean). This figure shows that we reach similar accuracy as the model of Pei et al. with as little as around 50 data points. Accuracy is only one measure that, in some cases, fails to detect issues with the method if the data sets are imbalanced. To test for those cases, we generally use the F_1 and κ tests. In this particular case, the data set is well-balanced, so these tests give similar conclusions as the accuracy. In the Appendix, we show how GPT-3 can also be used to accurately predict which phase (fcc, bcc, hcp) will form.

to compare machine learning approaches of material properties (i.e., standardized data sets and performance evaluation).

Table 4 compares the performance of a fine-tuned GPT-3 model with baselines. For molecules, we looked at properties ranging from highest occupied molecular orbital (HOMO)-lowest unoccupied molecular orbital (LUMO) gaps and solubility in water to the performance in organic photovoltaics. For materials, we focused on the properties of alloys, metal-organic frameworks, and polymers. And finally, for reactions, we considered two key cross-coupling reactions in organic chemistry. We compare the learning curves in the Appendix for each of these. Table 4 shows that in the low data regime, our GPT-3 model is typically at least as good as the conventional ML model and often needs fewer data. In the high-data regime, the conventional ML

Table 4: Data-efficiency comparison of best-performing GPT-3-based approaches with best-performing baselines. Numbers greater than 1 indicate GPT-3 is more data efficient. For the best comparison, we also split into (pre-trained) deep-learning (DL)-based baselines (here, MolCLR,⁴⁰² ModNet,⁴⁰³ CrabNet,⁴⁰¹ and TabPFN⁴⁰⁴) and baselines not using (pre-trained) deep-learning approaches (GPR, XGBoost, random forest (RF), automated machine learning optimized for materials science²¹²) on hand-tuned feature sets. There are several caveats to this analysis. First, focusing on the low-data regime might not always be the most relevant perspective. Second, we only focus on the binary classification setting in this table. Third, we focus on the F_1 macro score for this table (all cases are class-balanced). Fourth, we consider the performance of the GPT-3 model for ten training data points as a reference. We provide more details in Appendix F.6. The version of GPT-3 we utilized in this work has been trained on data up to Oct 2019 that mostly comes from web scraping (Common Crawl⁴⁰⁵ and WebText⁴⁰⁶) along with books corpora and Wikipedia. Structured datasets, however, have not been part of the training.

group	benchmark	publication year	best non-DL	best DL baseline
molecules	photoswitch transition wavelength ¹⁴³	2022	1.8	1.2
	free energy of solvation ⁴⁰⁷	2014	3.1	1.3
	solubility ⁴⁰⁸⁻⁴¹⁰	2004	1.0	0.02
	lipophilicity ^{411,412}	2012	4.9	0.97
	HOMO-LUMO gap ^{413,414}	2022	4.3	0.62
	OPV PCE ⁴¹⁵	2018	2.3	0.76
materials	surfactant free energy of adsorption ²⁰⁷	2021	1.4	0.37
	CO ₂ Henry coefficients ¹³²	2020	1.1	12
	CH ₄ Henry coefficients ¹³²	2020	1.0	0.59
	heat capacity ¹²⁵	2022	0.24	0.76
	HEA phase ⁴⁰⁰	2020	24	9.0
	bulk metallic glass formation ability ^{212,416}	2006	0.98	0.62
	metallic behavior ^{212,417}	2018	0.52	0.46
reactions	C-N cross-coupling ⁴¹⁸	2018	2.9	
	C-C cross-coupling ⁴¹⁹	2022	1.5	

models often catch up with the GPT-3 model. This makes sense as for a given size of the data set, the need for additional data and correlations captured by GPT-3 is less needed.

We have to mention that we did not optimize the fine-tuning of the GPT-3 model, i.e., we did not try to optimize how a sentence is presented to the model; one can envision that for chemical sentences, specific groupings (i.e., tokenization) can have better results.^{396,420,421} Also, we did not tune the number of times we show an example to a model (i.e., the number of epochs or the learning rate). The conventional models, on the other hand, have typically been optimized. Importantly, we are also not limited to fine-tuning; in Appendix F.5 we show that we can even achieve good performance *without fine-tuning* by incorporating examples directly into the prompt (so-called in-context learning,^{393,422} i.e., learning during inference time).

An interesting question is how to represent a molecule or material. Most of the literature uses IUPAC names. For ML applications, there has been a lot of effort to represent a chemical with unique line encodings (e.g., simplified molecular-input line-entry system (SMILES)⁴²³ or self-referencing embedded strings (SEFLIES)^{424,425}). As the GPT-3 model has been developed using natural language, one might expect that chemical names are preferred over line representations such as SMILES or SEFLIES. Therefore, we investigated different representations for our molecular property prediction tasks (see also Appendix F.4). Surprisingly, our results (see Appendix F.6) show that good results are obtained irrespective of the representation used. This

suggests that the GPT-3 model can map the different representations of a molecule to a similar internal representation. The fact that we often get the best performance using the IUPAC name of the molecule makes fine-tuning GPT-3 for a particular application relatively simple for non-specialists.

A more challenging task than classification is to make a regression model, which would allow us to predict the value of a continuous property such as the Henry coefficient for the adsorption of a gas in a porous material. As we are using a language model, an actual regression model that predicts real numbers ($\in \mathbb{R}$) is impossible (without changes to the model architecture and training procedure). However, in most, if not all, practical applications, the accuracy for which we can make predictions is always limited. For example, for the Henry coefficient of a material, an accuracy of 1 % (or a certain number of decimal points) is sufficient for most applications. Hence, we use molecules with Henry coefficients rounded to this accuracy as a training set and assume that the GPT-3 model can interpolate these numbers. Of course, one could also convert this into a classification problem by making very small bins and giving these a name. For this more challenging regression task, we need more data for tuning the GPT-3 model, and we still get a performance that can approach the state-of-the-art, but as this approach needs much more data, the advantage, except for the ease of training, is less. A similar conclusion we obtain for other, unrelated regression problems (see Appendix F.7).

6.1.1 Inverse design

One can argue that the ultimate goal of machine learning in chemistry is to create a model that can generate molecules with a desired set of properties. This is also known as inverse design.⁴²⁶ Broadly speaking, there are two approaches. If we have large datasets, we can train generative models such as variational autoencoders (VAEs)^{427,428} or generative adversarial neural networks (GANs).^{191,429} Without large datasets, evolutionary techniques such as genetic algorithms can generate novel, potentially interesting molecules.^{429–432} Those evolutionary methods work best if one can limit the underlying chemistry; for example, finding the optimal functional group on a material of which the backbone is well-defined.⁴³³

Given that the GPT-3 model can predict the properties of molecules and materials with a surprisingly small data set, trying an inverse design strategy is tempting. This would be particularly important in the early stages of research; one often has a small set of experimental data points and a limited understanding. Yet, we could leverage a fine-tuned GPT-3 model to generate suggestions for novel materials with similar or even better performance. This would be an incredible step forward. In particular, as we have shown, the tuning of such a natural language model is much more accessible than the training of conventional ML models. Here, we investigate this setting: Can a fine-tuned GPT-3 propose valid molecules and materials that satisfy the constraints or desired properties specified in a prompt in natural language? Again, we are illustrating the potential for a few case studies.

Molecular photoswitches are organic molecules with extended aromatic systems that make them responsive to light. Upon radiation, they switch reversibly between different isomers (with different properties, such as dipole moments). This reversible switching makes them interesting for applications ranging from sensing to drug discovery. These molecules are complex, making sufficiently accurate predictions using first-principles theory very expensive. Yet it is important to have some guidance to identify promising molecules, and machine learning models have been developed for this. One of the important properties of these photoswitches is the wavelength at which there is a maximum in the adsorption spectrum for the *E* and *Z* isomer. Hence, we fine-tuned GPT-3 with the same data used by Griffiths et al.¹⁴³. As we

have shown above, we can fine-tune GPT-3 to accurately answer questions like What is the pi-pi* and transition wavelength of CN1C(/N = N/C2 = CC = CC = C2) = C(C)C = C1C?".

For GPT-3, inverse design is as simple as training the model with question and completion reversed. That is, answer the question What is a photoswitch with transition wavelengths of 324.0 nm and 442 nm, respectively with a text completion that should be a SMILES string that is a meaningful molecule. This approach should be contrasted with the approach used by Griffiths et al.¹⁴³, in which a library of molecules is generated, and their ML model is used to evaluate the transition wavelengths of each material. If one has a lot of knowledge about the system, one can design large specific libraries that contain many promising molecules, including molecules with transition wavelengths of 324.0 nm and 442 nm. But, such a brute force technique is not what we understand as inverse design, as it, by definition, cannot predict a molecule we did not include in our library.

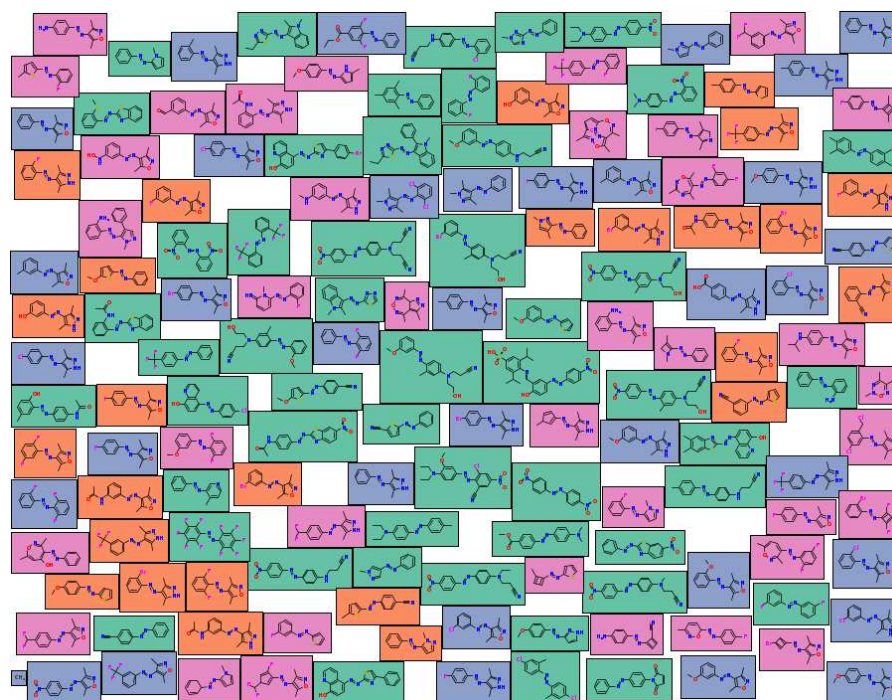


Figure 40: Molecule Cloud for randomly generated photoswitch molecules. Molecule Cloud generated using the tool reported by Ertl and Rohde⁴³⁴. Red background indicates samples from molecules in the database reported by Griffiths et al.¹⁴³ that our model did not generate, blue indicates the molecules our model generated and that are part of Griffiths's database, green background indicates samples that are generated by our model and that are not part of the database of Griffiths et al.¹⁴³ but part of the PubChem database. The purple background indicates molecules that our model generated but that are part neither of PubChem nor the database of Griffiths et al.¹⁴³

A simple test to see if our model can generate new structures is to ask it to generate molecules with transition wavelengths similar to those from the dataset reported by Griffiths et al.¹⁴³ Figure 40 shows a representative sample of the molecules generated by the model. As expected, many molecules come from the training set (colored orange in the figure). Importantly, many molecules are not in the training set, and interestingly, some are not even in the PubChem database of known chemicals. In Figure 41, we show that for the molecules, the transition wavelength is within a mean absolute percentage error of around 10 %.

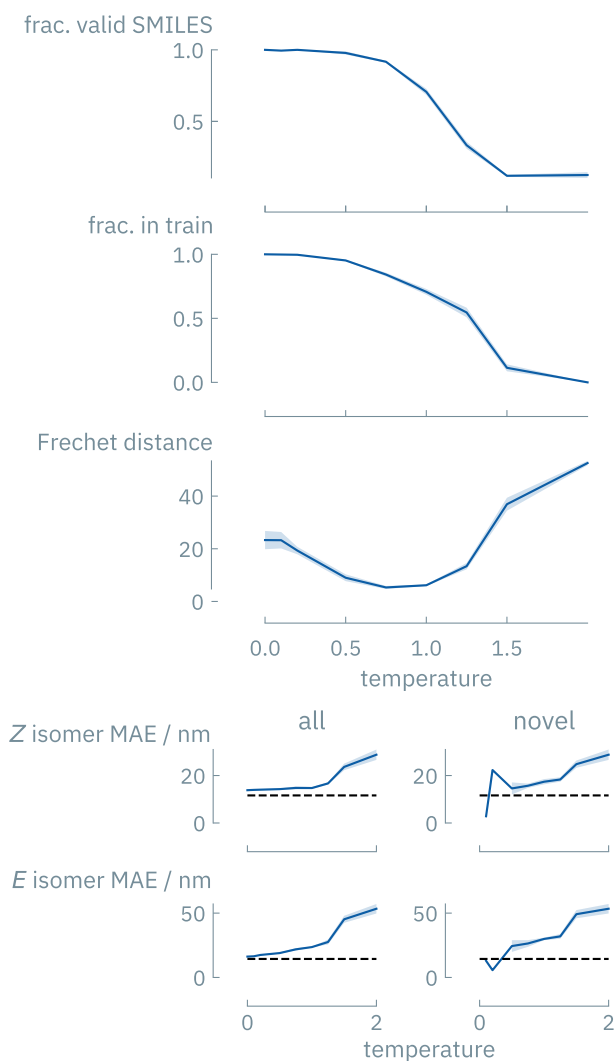


Figure 41: Photoswitch inverse design metrics as a function of temperature. The fraction of valid SMILES indicates the fraction of generated SMILES that can successfully be parsed using RDKit.¹⁷¹ We then determine the fraction (frac.) of those that already have been part of the training set and find that at low temperature GPT-3 tends to simply remember molecules from the training set. To quantitatively capture the similarity of the distribution of the generated molecules to the ones from the training set, we compute the Fréchet ChemNet Distance,⁴³⁵ which quantifies both diversity and distribution match¹³⁹ and goes through a minimum at intermediate temperatures. For quantifying how well the generated molecules match the desired transition wavelengths, we use the models reported by Griffiths et al.¹⁴³ to predict the transition wavelengths. In the figure, the dashed horizontal lines indicate the mean absolute error (MAE) of those models. Across all temperatures, we found high mean synthesizability (SA score⁴³⁶ smaller 3).

It is interesting to quantify how novel our newly generated molecules are. For this, we compare these molecules with the ones that Griffiths et al.¹⁴³ collected, one of the largest databases of synthesized azo-photoswitches. We quantify the similarity by computing the distance between molecular fingerprints. Figure 42 visualizes this by laying out the resulting (approximate) nearest-neighbor graph in two dimensions. The orange and green spheres represent molecules from the Griffiths dataset, the blue spheres show the novel ones, and the pink ones are not even part of the

PubChem database (the largest open-source chemistry database). As expected, we find many new structures that are derivatives of molecules in the Griffiths database. However, we also find branches that are not part of the library of Griffiths et al.¹⁴³, illustrating that we truly have carried out inverse design.

In generating these molecules, we adjusted the so-called softmax temperature in the GPT-3 settings. This temperature has been introduced to generate more natural text. If we set this temperature to zero, we will generate text with the most frequently used words. To make the text more natural, we can increase the temperature, making it more likely that less commonly used synonyms are chosen. For chemistry, if we aim to complete a SMILES starting with carbon, the zero-temperature solution would always complete the symbol that most commonly follows carbon ("(" in the QMugs dataset), whereas too-high temperatures would randomly choose any element. Hence we need to find a balance between the obvious and impossible chemistry.

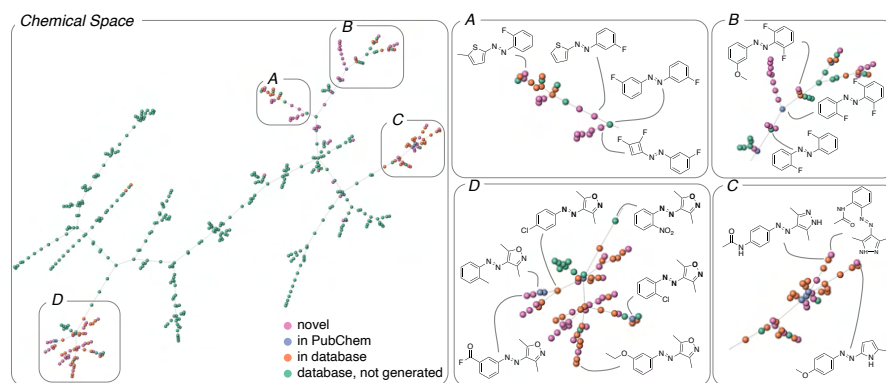


Figure 42: TMAP visualization of the generated photoswitches and the training set. The tree-map (TMAP) algorithm builds a nearest-neighbor graph, which is then embedded in two dimensions. Therefore, similar molecules are connected with an edge. We color the points depending on if they are part of the original dataset of Griffiths et al.¹⁴³ but not generated (green), part of the database, and generated by our model (orange). Our models can also generate molecules that have not been part of the photoswitch dataset (note that the model was only trained on 92 molecules from this database). In some cases, those molecules have been reported before and are part of the PubChem database (blue) or are not even part of PubChem (pink). From this figure, we see that the generated molecules sometimes substitutions for molecules in the dataset. In other cases, newly generated molecules introduce a completely new scaffold. For this visualization, we used the TMAP⁴³⁷ algorithm on photoswitch molecules described using MinHash fingerprint with 2048 permutations.⁴³⁸

The impact of the temperature parameter is shown in Figure 41. At low temperatures, the generated molecules often come from the training set and only show a low diversity. Across all temperatures, the generated molecules seem synthesizable, as judged by a low SA score.⁴³⁶ Increasing the temperature gives us more diverse and novel structures, but one can also expect more structures that make no chemical sense, i.e., are invalid.

6.1.2 Stretching the limits

The results on the photoswitches illustrate the potential of GPT-3 models for chemistry. As these models require very little knowledge of chemistry, it raises the question if the results can be trusted. Of course, one can always carry out the standard

machine-learning validation by splitting the dataset into a training and a test set, and this should give a clear indication of whether the GPT-3 model is better than a random guess. To get some more insights into why we can trust these GPT-3 predictions, we carried out some experiments where we tried to stretch the limits.

We have already seen that we can obtain good results independent of how we represent a molecule (IUPAC names, SMILES, or SEFLIES), but can GPT-3 interpret an abstract representation of molecules we invented? Jablonka et al.²⁰⁷ developed an active learning approach to design dispersants using a coarse-grained approach. This dispersant was a linear copolymer with four monomer types and a chain length between 16 and 48 units, giving a chemical design space of 58 million different dispersants. One important goal in this work was to find dispersants with the right binding free energy, i.e., which polymer length and which monomer sequence is optimal. As there is no way the GPT-3 knows about the properties or representations of the coarse-grained polymers, it is interesting to see if we can get any sensible result if we ask the question *What is the adsorption free energy of coarse-grained dispersant AAAABBBBDDDDAAAACCCC* or as inverse design, *Give me a structure of a coarse-grained dispersant with a free energy of 17*. Surprisingly, for the prediction of the adsorption free energy, the GPT-3 model outperforms the models developed by Jablonka et al.²⁰⁷ Additionally, it can also successfully carry out the inverse design and generate monomer sequences that give the desired composition and, with a mean percentage error of around 22 %, the desired adsorption free energy (the ground truth already has a mean percentage error of around 9 %, see Appendix F.8.1 for details). This example sheds some light on the power of fine-tuning of GPT-3. The essence of our free energy question is correlating a pattern of a small set of tokens to a free energy. To generate meaningful text, GPT-3 is apt at extracting patterns in text. GPT-3 can find that patterns of tokens are correlated to a property. It will learn that, for instance, our ABAA pattern can be correlated to the conventional notation of block-copolymers.

In the case of the photoswitches, we have seen that the GPT-3 model can generate new molecules that are quite different from the training set. To explore in detail how far we can stretch the limits of what new molecules we can generate, we choose an application for which quantum calculations are known to predict the experimental values sufficiently accurately. The HOMO-LUMO gap is such an application. For instance, the HOMO-LUMO gap is relevant in electronic applications that aim to excite a molecule at a specific energy. This HOMO-LUMO gap can be predicted accurately using semi-empirical quantum mechanics (GFN2-extended tight-binding (xTB)⁴³⁹), which is computationally affordable enough for us to compute for all generated molecules (see Figure 239). Moreover, the QMugs dataset,^{413,414} has listed these HOMO-LUMO calculations for 665 k molecules. Here we use the quantum calculations as the “ground truth” to validate our predictions.

In the Appendix, we show that with the training of only 500 samples, we can get a reasonable estimate of the HOMO-LUMO gap of the molecules in the QMugs dataset. Also, by reverting the question, we have our model trained for inverse design. In Appendix F.8.3, we show that by asking the model *What is a molecule with a HOMO-LUMO gap of 3.5 eV*, we get similar to the photoswitches, a set of promising novel molecules. These novel molecules are not part of our training set and not even part of the QMugs dataset.

We now conduct some experiments to test how well the GPT-3 model can extrapolate to HOMO-LUMO gaps for which it has not received any training. To mimic this situation, we retrained our inverse design model using a dataset that only has molecules with HOMO-LUMO gaps smaller than 3.5 eV and subsequently query the model with a question that requires the GPT-3 model to extrapolate. We do this by asking 1,000 times the question: *What is a molecule with a HOMO LUMO*

gap of $\langle XX \rangle$, where each time we slightly change the value of the HOMO LUMO gap, i.e., we sample XX from a Gaussian centered at 4 eV. Interestingly, the GPT-3 model does provide structures with a distribution of which our quantum calculations confirm that a significant fraction has a HOMO-LUMO gap > 4.0 eV. Again this is a remarkable result. In our training set, there was not a single molecule with a band gap > 3.5 eV, which shows that the GPT-3 model can make useful extrapolations. We can do a similar experiment for the photoswitches, for which we might have a library of photoswitches whose transition wavelengths are all below 350 nm. For practical applications, however, it can often be essential to have adsorption at larger wavelengths. In this case, we can successfully use a fine-tuned GPT-3 model to generate photoswitch molecules that adsorb at lower energy (Figure 237, which we also validated with time-dependent DFT (TDDFT) in Appendix F.8.2).

These findings inspired us to do an inverse design experiment, aiming to design molecules with properties that take us very far from the training set.⁴⁴⁰ We are interested in molecules that have a HOMO-LUMO gap > 5 eV. From the distribution of HOMO-LUMO gaps in the QMugs database (see Figure 43), we see that the average band gap is around 2.58 eV. There is only a hand full of molecules that have a HOMO-LUMO gap above 5 eV in this database. Hence, this is a challenging inverse design problem; there are too few materials in the database that have the desired properties; hence conventional machine learning can give us little if any, guidance. Here our experiment is the quantum calculation, and we typically assume that we can evaluate hundreds of materials in a reasonable time. If we use fewer evaluations, the model struggles to generate molecules with HOMO-LUMO gaps beyond the initial distribution. From a machine-learning point of view, a set of hundreds of materials is in a very low data regime. However, from an experimental point of view, this is a significant but doable effort. Of course, this is a somewhat arbitrary limit, and in Figure 245 we also give data for significantly fewer experiments.

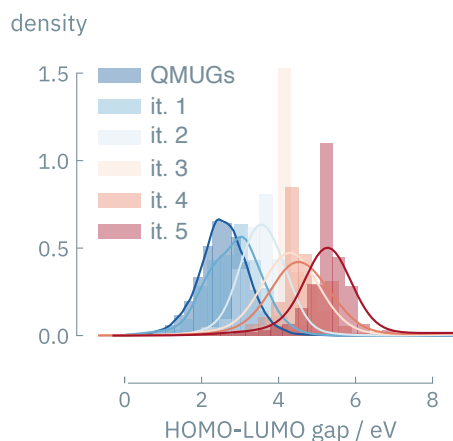


Figure 43: Iteratively biased generation of molecules toward large HOMO-LUMO gaps using GPT-3 fine-tuned on the QMugs dataset of draws. We start by fine-tuning GPT-3 on a sample of the QMugs dataset and use this model to query for gaps from a normal distribution with shifted mean (mean 4.0 eV, standard deviation 0.2 eV). We then iteratively select the high-gap samples of the generated molecules and fine-tune the model on this data (i.e., starting from the second generation, the model is fine-tuned on molecules it itself generated). Smooth curves show kernel-density estimates; the plot is truncated at 10 eV, but the models also generate some molecules with larger HOMO-LUMO gaps. If we limit the number of quantum chemistry evaluations to 100, we can still successfully shift the distribution as shown in Figure 245.

We start with the training using a set of hundreds of molecules randomly selected from the QMugs dataset (blue distribution in Figure 43). These selected molecules will have band gap distribution similar to the QMugs dataset. We then query for HOMO-LUMO gaps, now around 1000 times requesting a molecule with a band gap taken from a normal distribution with shifted mean (mean 4 eV standard deviation 0.2 eV). We evaluated these new molecules (green curve in Figure 43), which indeed shows a shift of the distribution to higher HOMO-LUMO gaps. In the next iteration, we retrain the model with the new data and query again higher HOMO-LUMO gaps. Figure 43 shows that we have achieved our aim after four iterations.

In many practical applications, one has more requirements than the correct HOMO-LUMO gap. Suppose we need a bromine-containing material with a well-defined HOMO-LUMO gap. Without additional training, we queried our model using `What is a molecule with a HOMO-LUMO gap of 3.5 eV and Br as part of the molecule.` Figure 44 shows the results of such questions for different functional groups. We do not necessarily generate more molecules with the desired functional groups if we take the low-temperature results. However, at higher temperatures, where we allow for more creativity, we generate molecules with as much as ten times more structures with the desired functional group than the query without specifying the functional groups. As we did not train the model for this type of question, it is not obvious whether it can recognize Br as an element and add it to a chemically meaningful place in the SMILES string. Yet it is known that these models can give meaningful answers without any training (zero-shot). It is fascinating that the GPT-3 model can connect a SMILES string and the part of the query that Br should be part of the molecule.

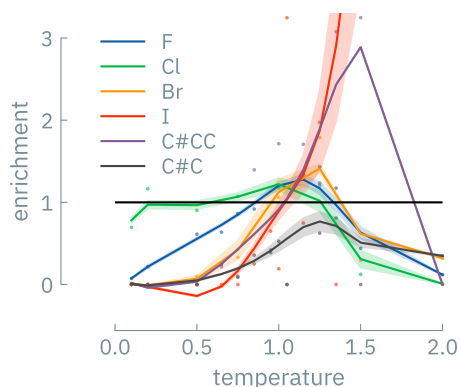


Figure 44: Generating molecules with desired functional groups. The figure shows the enrichment, i.e., the ratio of the fraction of functional groups in the generated molecules to their occurrence in QMugs for different functional groups as a function of temperature (see Appendix F.11). The first entries in the legend show the simple addition of halogens. The last two show the addition of more complex functionalities (alkyne, methyl-alkyne). For all points above the horizontal line, the generated molecules contain the desired functional group more frequently than in the QMugs distribution. For the smooth curves, we performed local polynomial regression (Gaussian kernel of width 0.25 and degree 2). Original data points are shown with dots.

It is interesting to ask our fine-tuned models query that does not make any chemical sense, e.g., `What is the transition wavelength of Berend?`. Table 5 shows some of the answers we get. For those tests, the models always generated the same class. Also, for the inverse design models, we often obtain the same output molecule (Appendix F.10). GPT-3 may convincingly hallucinate an answer for a

clearly invalid input. For these cases, the strength of GPT-3, that it can learn chemistry from many different inputs, is also its main weakness; it does not have a filter for questions that do not make any chemical sense, and we also have no guarantees that the answers make chemical sense. In conventional machine learning approaches, one has to convert inputs into a feature vector and, in this stage, filter out chemical nonsense (such as inputs that are not molecules).

Table 5: Completions to queries that are invalid or do not have any chemical meaning. For this experiment, we used a model we fine-tuned to predict the transition wavelength of photoswitch molecules and replaced the molecular representations with various other strings.

prompt	completion
What is the transition wavelength of Berend?	0
What is the transition wavelength of Kevin?	0
What is the transition wavelength of Philippe?	0
What is the transition wavelength of Andres?	0
What is the transition wavelength of Bus?	0
What is the transition wavelength of car?	0
What is the transition wavelength of tree?	0
What is the transition wavelength of house?	0
What is the transition wavelength of cat?	0
What is the transition wavelength of magnificent?	0
what is the adsorption energy of Berend?	0
what is the adsorption energy of Kevin?	0
what is the adsorption energy of Philippe?	0
what is the adsorption energy of Andres?	0
what is the adsorption energy of Bus?	0
what is the adsorption energy of car?	0
what is the adsorption energy of tree?	0
What is the transition wavelength of OObZnK?	0
What is the transition wavelength of ZnSiZnFeMn?	0
What is the transition wavelength of CaBeHNBe?	0
What is the transition wavelength of NiHNeFeS?	0
What is the transition wavelength of MnNiNiCoO?	0
What is the transition wavelength of ZnAsMgZnNi?	0
What is the transition wavelength of lfplx?	0
What is the transition wavelength of lvdzu?	0
What is the transition wavelength of hvdos?	0
What is the transition wavelength of bdxsu?	0
What is the transition wavelength of mhokz?	0
What is the transition wavelength of padfx?	0
What is the transition wavelength of gdhpr?	0

6.1.3 Concluding remarks

Our results raise a very important question, how is it possible that a natural language model with no prior training in chemistry outperforms dedicated machine-learning models? To our knowledge, this fundamental question has no rigorous answer. The fact that we get good results independent of the chemical representation illustrates that these language models are very apt at extracting correlations from any text. For example, we found good results using both conventional chemical names and completely hypothetical representations. In both cases, the model could quantitatively correlate the pattern of repeating units correctly to different kinds of properties. In some regards, this is not that different from how an experienced chemist would design a material. Suppose a chemist sees a publication of a new material, and she or he notices that some properties are very similar to the materials studied for a completely different application. These similarities are often the source of inspiration to

try variations of this novel material for this application. Extracting and remembering such correlations enables GPT-3 to perform (inverse) design. That GPT-3 can work with these correlations over such a large range of topics makes it so powerful.

Of course, we would like to emphasize that if we say that our GPT-3 model is successful, it only implies that we have established that our GPT-3 model has identified correlations in the current training data that can be successfully exploited to make predictions. However, this does not imply that the correlations are always meaningful or related to cause-effect relationships. Hence, our research does not stop here. Using GPT-3 to identify these correlations and ultimately get a deeper understanding will be the next step. In this context, we argue that GPT-3 is only a tool to make more effective use of the knowledge scientists have collected over the years. At this point, it is also important to mention that most scientific literature (including all failed or partially successful experiments³⁸) has not been seen by GPT-3. Hence, one can expect an even more impressive performance if this literature is added to the training data.

As we show in this work, a machine learning system built using GPT-3 works impressively well for a wide range of questions in chemistry—even for those for which we cannot use conventional line representations such as SMILES. Compared to conventional machine learning, it has many advantages. GPT-3 can be used for many different applications. Each application uses the same approach, in which the training and use of the model are based on questions formulated in natural language. This raises the bar for future machine learning studies, as any new models should at least outperform this simple approach instead.

The other crucial practical point is that using a GPT-3 model in a research setting is similar to a literature search. It will allow chemists to very effectively leverage the chemical knowledge we have collected. GPT-3 has been designed to discover correlations in text fragments, and the fact that these correlations are extremely relevant to chemistry opens many possibilities for chemists and material scientists alike.

DATA AVAILABILITY

All data used in this work was obtained from public sources and can be downloaded with our Python code (<https://github.com/kjappelbaum/gptchem>).

CODE AVAILABILITY

All code created in this work is available on GitHub (<https://github.com/kjappelbaum/gptchem>).

7

CONCLUSION AND FUTURE
RESEARCH

7.1 CONCLUSIONS

Material design is an optimization problem across multiple scales where we often lack tools or theory for systematic design—due to both reducible and irreducible complexity.²⁶ The works presented in this thesis show that data-driven approaches can be used in the material discovery process across all relevant scales: From the prediction of oxidation states on the atom scale over adsorption properties on the mesoscale to the prediction of solvent emissions on the pilot-plant scale. In those cases, ML could make predictions possible or faster or act as a muse⁴⁴¹—for example, by inspiring engineers to perform further experiments on their plant.

However, the previous chapters highlighted many pitfalls and challenges for data-intensive research approaches, which the efforts in this thesis address.

A crucial limitation of most, if not all, the ML models built as part of the thesis is that they cannot yet provide actionable insights for a chemist or chemical engineer—who would like to know which compound they can make in the lab (at scale). This problem is not unique to ML (classical computational chemistry also faces this challenge). However, ML seems like an ideal fit for addressing this problem: The questions of synthesizability are too complex to address with theory but might be answered from data. The efforts around developing data capture systems for chemistry might pave the way toward also addressing those questions (Chapter 1).

With the developments presented in Chapter 2, we provide digital (reticular) chemists with a toolbox to leverage this data. While we showed that previous efforts might not have achieved the ultimate impact due to suboptimal model evaluation practices—we now have the infrastructure to avoid this. In particular, we also provide tooling—in a unified framework—to build highly predictive representations of materials. While we have been using the tools described in Chapter 2 to build highly predictive models for properties such as colors,³⁴ oxidation states,¹²² band gaps,¹¹⁴ and gas uptakes,^{114,132} this should only be the beginning. Applying these tools to more properties, such as photocatalytic descriptors or carbon capture process performance metrics, is a natural next step. However, all these models will only be of use if chemists can use them. For this, they need to be made easily accessible and provide explanations and uncertainty estimates.

We cannot expect those models to be perfect for all cases. Therefore it is important also to collect feedback—on the predictions, but also the explanations—and incorporate this as part of an active learning loop since it is not the performance on a test set (drawn from the same distribution as the training data) that matters but the one in a real-world application.

For an active learning application in material science, it is essential to properly deal with the fact that material design is multiobjective. In Chapter 3, we presented an algorithm that can do so and has now found various applications beyond the original polymer design task. However, there are still challenges. First, similar to many other active learning and Bayesian optimization frameworks, the algorithm presented in Chapter 3 is sensitive to the choice of hyperparameters. In developing the algorithm on historic datasets, this is not necessarily a problem, as one can easily test different parameter settings. However, in practice, one needs to choose specific parameters *prior to* starting an experimental campaign. One possibility to address this challenge

is to use ensembles over different parameter settings or to meta-learn⁴⁴² good hyperparameter settings on related datasets. In addition, the current algorithm operates on a fixed design space and hence requires that relevant materials are enumerated. Chemical space is, however, so large that enumeration is impractical. Hence, a natural extension for future work is to use generative models to adaptively enumerate relevant parts of the chemical space. However, these efforts are also only useful if they are accessible to synthetic chemists and material scientists.

Chapter 6 presented an approach that can make machine learning more accessible: By consolidating different tasks in a convenient natural-language-based framework. We showed that fine-tuning or in-context learning of GPT-3 can give surprisingly good performance on classification, regression, and inverse design from molecules over materials to reactions. These results, along with the findings from White and co-workers,^{394,395} show the transformative potential of foundation models in chemistry.

7.2 FUTURE RESEARCH

LLMS IN CHEMISTRY One can make two undeniable improvements on the work presented in Chapter 6: First, building a chemistry-specific dataset and second, using chemistry-specific tokenization (which has been shown to be of high performance in prior work such as the Galactica model).³⁹⁶ To address these challenges, large computational resources are required that are not easily accessible to labs in academia. As a core team member of the ChemNLP project, we can leverage the massive computational resources provided by stability.ai to collect an extensive chemistry dataset and build a foundational model for chemistry. The potential uses of this model are ample: Much of chemical information is hidden in text form—a foundational model for chemistry can aid in converting it into machine-actionable form.³⁹⁷ As a specific step in this direction, we are currently using the file parsers developed as part of the data infrastructure we developed (Chapter 1 and Chapter 2) to develop a dataset tailored for building models that can write parser code. Beyond that more technical use of foundational models, there is also a unique possibility that the natural language setting provides: For many applications, such as predicting reaction outcomes, the context, e.g., how the reaction was carried out, is essential. In many cases, however, this context can only be fully described in text form. LLMs could provide us with an avenue to also consider this context while linking it to other parts of chemistry. This latter point is also significant on its own: By training the models on a large body of knowledge, we could provide every chemist with a virtual assistant—a patient, experienced chemist that has read all the literature. However, it is also important to realize that only focusing on the text modality is not enough—much of chemical data is better represented in other forms (images, graphs), and hence, we must work on making foundation models multimodal.

SCIENTIFIC UNDERSTANDING AND ROBUSTNESS In contrast to a real chemist, however, those models will also answer if they have no clue. And those models will also provide no explanations for their reasoning—even though scientific understanding is an important part of doing science. And we can only expect a model that is right for the right reasons to generalize to a new dataset. Future work must focus on making ML more focussed on mechanistic understanding and not the minimization of a loss on a training set. To make this possible, interpretable and chemically meaningful representations are needed. A graph-based representation might seem intuitive for chemical compounds (chemists naturally think in terms of atoms and bonds). However, it is not sufficient alone to address the multi-scale

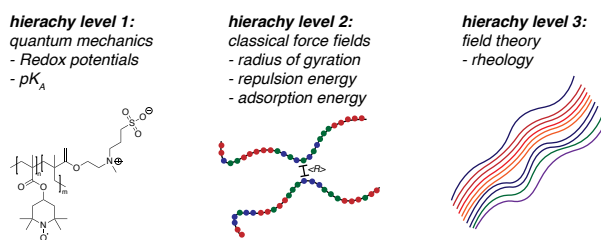


Figure 45: Hierarchy is an important architecture principle for natural and engineered systems. Machine learning must understand this building principle to provide actionable and meaningful insights across length and time scales. For polymers for energy applications that requires, for instance, the model to understand the relationships between, atom, repeating unit, and polymer aggregates on the macroscale—analysis of each of which alone will not generate enough insight for the design of a polymer that works in the real world.⁴⁴³

nature of material design. For instance, for polymer materials, the model should be intelligible not only on the level of atoms and bonds—but also on the level of monomers or blocks of building blocks.

CLOSING THE LOOP AND BRIDGING THE SCALES Importantly, such an improved representation might help us bridge the scales. While this thesis has used ML on vastly different scales, it has not yet bridged them. To a large extent, this is because models operating on different scales use very different representations as input. The community would benefit tremendously if approaches were developed that can work across scales—for instance, by (learnable) adaptive pooling in graph neural networks.⁴⁴⁴ Doing this might also bring along technical advances, such as addressing the problem that it is challenging to exchange information between distant atoms in a message-passing neural network (NN) (MPNN).

Ultimately, also bridging the scales is not enough. We need to provide chemists—or self-driving labs—with actionable insights. That is, our ML systems need to also consider constraints from the consideration of life-cycle analysis (LCA) and planetary boundaries as well as synthesizability, scalability, and economics. Clearly, this requires breaking the walls of traditional chemistry departments and genuinely embracing the transdisciplinary nature of the central science.

EMBRACING THE CENTRAL SCIENCE Chemistry might be the grand application of ML: Chemistry helps us cure diseases, fight climate change, and can provide us with an enormous growth of quality of life. Many questions in chemistry are too complex or too expensive for conventional approaches. In some subfields, ML has already revolutionized the research practice. Progress in others, however, is still bottlenecked by collective action problems, traditions, and suboptimal research practices. In addition, we still struggle to leverage all the unique features of the central science: We have many experienced chemists that could help instill our models more chemical reasoning by closely and routinely interacting with them. Domain scientists have been extracting and re-discovering patterns for a long time: For heterogeneous catalysis, homogeneous catalysis, biocatalysis, and beyond. However, all this data and knowledge is still stuck in silos and not used as an inductive bias for our models. For instance, why don't we “bootstrap” our models or Bayesian optimization runs with closely related datasets? It is time to embrace the central science's multidisciplinary nature, break the walls of silos and leverage the best from all neighboring disciplines. Let's embrace the *central* science.

Part IV

APPENDIX

A

SUPPORTING INFORMATION FOR “MAKING THE COLLECTIVE KNOWLEDGE OF CHEMISTRY OPEN AND MACHINE ACTIONABLE”

A.1 IMPLEMENTING AND MAINTAINING AN OPEN SCIENCE INFRASTRUCTURE

In this part, we want to focus on the implementation, discussing some key lessons we learned from implementing an ELN. We think chemistry can learn several lessons from the world of open source.

A.1.1 Survival advantage of the worse

In a networked world, the worse solution can have an advantage over an over-engineered one (the infamous “worse is better” principle).⁴⁴⁵ Over the years, the chemistry community could observe the proposal of a welter of ontologies, schema, and ELNs. While those are admirable efforts, we feel that new file formats will not help the community. In particular, they will not address the biggest problem, which is the lack of interoperability between existing solutions. For most data types, there are already standardized, sometimes even IUPAC-recommended vocabularies, schema, and serialization formats, and from our experience, one cannot anticipate all eventualities of a data schema (chemistry is likely just too flexible, some even argue that a schema-first approach will never scale⁴⁴⁶). Many successful technologies experienced that after an initial design (that is good enough for a potential disruptive innovation), it is much more worthwhile to implement a prototype and keep on iterating based on user feedback (which will be conditioned on imperfect design, but stick to it as it provides them value). The “worse” solution will have better survival characteristics because it is easier to iterate on it (whereas an overly complicated solution might become impossible to maintain).

In this context, it is also interesting to reflect on what Oleksik et al.⁸¹ described as “a tension that is intrinsic to the digital nature of ELNs: a conflict between the flexibility, fluidity, and low threshold for modifying digital records and the requirement for persistence and consistency”. In the ELNs that are currently used in the chemistry community, one finds both extremes: Some interfaces are reminiscent of filling a long form (like for a tax declaration), whereas others provide no structure at all (like a simple note-taking app). From the viewpoint of data capture (seeing the data analysts as the end-users), a highly structured form might be “best”—but it will be a worse overall solution as chemists who are supposed to enter the data will not adopt it.

We find that a key design requirement is to make the barrier for data entry as low as possible, potentially using some ideas like predefined sentences with variable fields, to increase adoption. For chemistry, this also means that an ELNs should support the editing of chemical structures—also to be able to store this information in a reusable form and not as an image that was created by another software.

A.1.2 Software can be maintained and improved by the community

Some of our experimental colleagues raised the viewpoint that complicated pieces of software cannot be maintained by researchers.

Our everyday life shows that the opposite is true. We all use software that is developed and maintained by the community. Our world would, as we know it, would not be possible without open-source projects such as Linux, Firefox, or Python. All these pieces of software are probably better because they are open, hence better tested (“given enough eyeballs, all bugs are shallow”⁴⁴⁷). Clearly, there are issues with the sustainability of open-source software, and there are initiatives that call for novel funding schemes^{448,449}—but there are also many examples that show that successful businesses can be built around open software. For example, the SciNote ELN is licensed under the open Mozilla public license, but there is still a business model that sells deployments of the ELN on the cloud together with support—for users that do not want to deploy the ELN themselves. A similar scheme is used for the eLabFTW ELN where Deltablot sells “pro support” and hosting, i.e. service level agreements (SLAs).

On the other hand, we can think of an open science infrastructure as something like our NMR facility. Our departments have funds to keep this infrastructure running. It seems forward-looking to do the same for a data infrastructure that not only supports research but could also enable new research.

Notably, the open-source model has not only been proven successful for software development but also for the sciences. One example is the polymath project where the maths professor Timothy Gowers used his blog to ask for help with a proof for a central theorem.⁴⁵⁰ Within less than two months, the results—based on about 800 comments from 27 people—were being written up in a paper. Similarly, an open-source science project found a new pathway to an enantiopure form of the drug praziquantel.⁶⁵ One of the reasons this model works is that the relevant expertise and the new viewpoints can make themselves heard—no research group can have all the experience and expertise there is in the world, and often an overly specialized team can struggle to escape a local minimum in their thinking.^{65,451} The acceleration of research that both the polymath and the open-source malaria project observed was described by Timothy Gowers, the initiator of the polymath project, as “It feels as though this process is to normal research as driving is to pushing a car.”⁴⁵²

As with open-source software, open-source science is also interesting because the creation process is open. We can follow all the discussions that led to the development of Linux, we can follow all the small steps that have been made for the polymath project, and we can also see how the route to the enantiopure form of praziquantel was discovered. All this can be used as an educational resource, but also to understand why things are as they are.

An open science infrastructure like the one outlined in this work can be a key ingredient for this process—by simply opening the lab notebooks and allowing others to comment on it. One can even envision that (parts of) the data captured via the ELNs is fed directly as a dataset into the Kaggle platform,⁴⁵³ where (aspiring) data scientists can use it to develop new machine learning models.

A.1.3 Modularity, on different levels, is key

An ELN is a complex piece of software. If one were to develop it as one giant monolith, it would be infeasible to try to fix or update one part of the code without breaking another one, and it would also hamper parallel development on different parts of the codebase by different developers (“Linus does not scale”). A monolith is not scalable. Simply alone for this reason, it is vital that the tool is developed at the most

granular level. But besides making development and maintenance easier, it comes with other advantages. The world of (web) development is fast changing. No one knows what the new, state-of-the-art technologies will be in 10 years. By tightly coupling all parts of the ELN, it will be completely unfeasible to upgrade to a new framework. For example, if one keeps the frontend (the client side code that runs in the browser and visualizes the data) separate from the backend (the code that runs on the server and does the heavy lifting) one can much easier migrate to new frameworks.⁴⁵⁴ One new ELN framework that followed this approach is the Chemotion ELN developed at the Karlsruhe Institute for Technology (KIT). One advantage of developing the frontend in a modern web framework is that the resulting service can be used on any platform—also on mobile devices—via the web browser.

One challenge digital records face compared to paper-based notes is immediacy.⁶³ It is usually much faster to quickly jot something down on paper than to enter it via a clumsy interface on a computer that might not even be in the lab. For this reason, we envision that the ELN must improve how researchers can ingest data. One way we explored in our work is chatbots³⁴ that easily allow capturing pictures or videos—something that is typically much harder to do with paper-based notebooks and that would allow capturing the experiments in much more detail. Others explored to improve the ease of use using speech recognition,^{455,456} but these techniques did not find widespread use, one reason for which the specialized chemistry vocabulary might be.

The LabTrove ELN showcases an advantage digital solutions can have over conventional paper-based notebooks: They allow one to connect the reflection about the data directly with the data records themselves. In the context of LabTrove, one might write summary blog posts in preparation for weekly group seminars that link to particular experiments. In this way, not only the experiments themselves but also the reflection on the experiments and the thought processes that led to the next steps are captured—similar to what happens in open-source science projects.

A.1.4 The user base is heterogeneous

One challenge every ELN faces is that the user base is heterogeneous. On the one hand, this stems from the fact that, in our perspective, the users are not only the experimental chemists but also scientists that will re-purpose the data. Both favor different ways of interacting with the system. But also within the group of experimental chemists, there are varying amounts of interest in what is going on behind the scenes. On top of that, chemists will attempt to use the ELN from different platforms. Many would prefer to use it from a mobile device but access also from their Mac, Windows, or Linux desktop computer. One viable solution to address all these different needs is to offer the ELN as a cloud service that can be accessed from a web browser on any device. This also simplifies the rollout of fixes and updates, as the developers do not depend on users installing them. Lastly, this also simplifies users' life as they do not need to install anything—they just need to open their web browsers. Note that it can make sense to implement the ELN such that most operations take place on the client side. This is feasible as the most common operations in the ELN are not resource-intensive (e.g., looking up data in a database, plotting spectra, writing notes). It can also make the ELN more responsive than the design in which *all operations* take place on a server (e.g., implemented using Jupyter notebooks, where one would need to start one container per user and cannot profit from many of the optimization techniques that modern web frameworks offer).

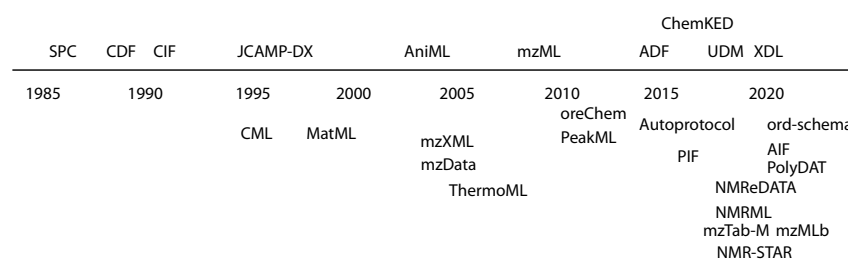


Figure 46: Non-comprehensive schematic timeline of standards proposed for chemical data. Inspired by Anderson et al.⁴⁶⁵.

A.1.5 Convergence and interoperability

For this vision to be sustainable on a larger scale, different solutions must be able to talk to each other. It is clear that there cannot be only one database or only one ELNs as the needs of different communities are just too diverse. But the different tools must be interoperable to make data reusable and harvestable on a larger scale. Standardization will also be key to integrating “manual chemistry” captured via ELNs with syntheses executed by robots and to link experimental work with computational works.

While there have been many efforts to create standards for data management in chemistry (some examples are described in Table 6 and 7), there are only a few (for example, the CIF) that are widely used. Motivated by this, some initiatives have tried defining standards for the field (see Table 8). However, some of the most advanced efforts are spearheaded by industry consortia, like the Allotrope Framework, which in some cases, requires membership to see the full documentation. While this can be a way to fund the efforts, it can also undermine one of the purposes for standardization, i.e., to make data reusable by everyone.⁴⁶⁶ We have to realize that one of the features of open source is that it can level the playing field, i.e., give less well-funded groups the same access to high-quality tools. Moreover, in the recent pandemic, we could witness how access to data can make a difference in response to an emergency.^{467,468}

We know from the computational materials science community that progress on open standards, even though slow, is possible. Starting in 2016, the OPTIMADE⁴⁶⁹ consortium, in which all major computational databases have been represented, agreed upon a specification with which the databases can “talk” to each other. Practically, this means that users can easily retrieve, combine, and compare data from all major databases.

Table 9: There is no lack of standards. This table is a non-comprehensive overview. The most common serialization language (serial.) is XML. XML is a web standard that is supported in most programming languages, is easily extendable, and is human-readable. It was originally developed as markup language and not as a data storage format. In contrast to XML, JSON is relatively easy to parse and is also supported by most, if not all, programming languages (often having a direct mapping to standard datatypes such as dictionaries). Self-defining Text Archive and Retrieval (STAR) is the file structuring used in CIF. Not human-readable are HDF5 (Hierarchical Data Format), netCDF (which in the most recent implementation builds on top of HDF5), SQLite, and protocol buffers. Those file types have the advantage that they can represent data more compactly than the text-based format. The reason for that is, for example, the numerical accuracy of numbers. In string-based representation, it takes more space (to “print” all the digits) than in a native representation. One difference between the standards in this table is the extent to which the formats provide controlled vocabularies. Some only provide basic implementations like `parameter` labels with which any parameter can be stored, whereas others specify a tightly controlled vocabulary that only allows specific terms (blurring the line between format and schema). Only a few reuse and extend vocabularies that are more widely used, e.g., the web in general (`schema.org`). This overview clearly reflects that there will not be a one-size-fits-all format, given that some applications need formats optimized for performance, whereas others will require audit trails and digital signature features. But the table shows that many implementations are not that different, e.g., multiple HDF5 implementations of the mzML standard, and that the community most likely does not need one additional (potentially backward incompatible) re-implementation of the mzML standard. What is interesting to observe is that most formats do not have a formalized mechanism for the proposal and discussion of changes to the standard. While some use public mailing groups and issue trackers, to our knowledge, only autoprotocol follows the schema of “enhancement proposals”, which is used in major software developments to propose and discuss new designs in a standardized, public-facing, and documented way. The inspection of the table also highlights that formats with buy-in from industry (e.g., ADF, UDM, AniML, Autoprotocol) found significantly wider adoption and are better maintained.

name	year	scopes	description	serial.
JCAMP-DX ⁴⁷¹	1988	general analytical chemistry	IUPAC recommended format. Many instruments can export this format, many databases such as the NIST web-book understand this format. Allows for some compression. Can be customized using private labels.	its own
ANDI ⁴⁷²	1992	general analytical chemistry	The Analytical Data Interchange format is most used in its flavor for mass spectrometry, ANDI-MS more widely used. Designed by ASTM	netCDF
CML	1995	chemical data in general	Is supported by many common programs and has been extended, for example, for spectra (CML-Spect). ⁴⁶⁴ One core idea of designers of the CML was that the information should be validatable, for example, to automatically find if there negative atom counts	XML
NMRSTAR ⁴⁷³	1996	(biological) NMR	archival format used by the Biological Nuclear Magnetic Resonance data Bank (BMRB), the international repository of biomolecular NMR data	STAR
matML ⁴⁷⁴	1999	materials property data	Development initiated at NIST with the goal to develop a format for the interchange of materials information, but did not find wide usage	XML
SpectroML ⁴⁷⁵	2002	UV-VIS	super-seeded by AniML	XML

ThermoML ⁴⁷⁶	2003	thermophysical and thermochemical property data	IUPAC standard for storage and exchange of experimental thermophysical and thermochemical property data. Data produced by the NIST Thermodynamics Research Center are provided in this format. IUPAC-based terminology is used as basis for data tagging	XML
GAML ⁴⁷⁶	2003	general analytical chemistry	The Generalized Analytical Markup Language attempted to make it easier to store multi-detector data (compared to JCAMP and ANDI). Explicitly avoids mapping vendor specific metadata to a common dictionary	XML
mzXML ⁴⁷⁷	2004	mass spectrometry	strict schema with enumerated attributes, officially deprecated in favour of mzML	XML
AnIML ⁴⁵⁷	2004	general analytical chemistry	The Analytical Information Markup Language aims to document workflows. It allows to embed digital signatures and audit trails, aims to be flexible enough for novel spectroscopic techniques. Found already support by some manufacturers like Agilent or ELNs such as LabWare. BSSN software (now owned by Merck) implemented converters for more than 150 instruments ⁴⁵⁸	XML
mzData ⁴⁷⁸	2006	mass spectrometry	Designed to be flexible via extendable controlled vocabulary	XML
mzML ⁴⁷⁹	2008	mass spectrometry	Aims to combine the best elements of mzXML and mzData, metadata is accurately and unambiguously annotated using the PSI-MS controlled vocabulary	XML
mz5 ⁴⁸⁰	2012	mass spectrometry	Uses HDF5 in an attempt to address performance issues with mzML. Manual mapping to the field labels of mzML	HDF5
ADF	2013	general analytical chemistry	The Allotrope Data Format features an ontology based on the widely used Basic Formal Ontology. Data is stored in RDF graph, can be programmatically validated. The format also supports audit trails. Additional files can be stored in virtual file system. Full analytical life-cycle can be stored in one file. Wide support from instrument manufacturers. Maintained by the Allotrope Foundation, which was founded by pharmaceutical companies, and which uses the membership fee paid by industrial members to contract external partners to develop and maintain the format	HDF5
mzDB ⁴⁸¹	2015	mass spectrometry	Attempts to optimise for high-throughput data processing and storing	SQLite
autoprotocol ⁴⁸²	2016	life science protocols	Autoprotocol is directly mappable to hardware commands for robotic automation. Notably, it has a mechanism for the proposal of changes to the standard (Autoprotocol Standard Changes). It is maintained by Strateos	JSON
PIF	2017	information about physical systems	The Physical Information File was designed to be able to describe a broad array of data “from parts in a car down to a single monomer in a polymer matrix”, hence can have a nesting of subsystems. It is maintained by Citrine Informatics	JSON
UDM	2017	experimental information about compound synthesis and testing	The Unified Data Model is a format to store reaction data (schema, conditions, provenance, etc.) and biological testing. Has been maintained by Elsevier, who transferred the ownership to the Pistoia Alliance.	XML

NMReDATA ⁴⁸³	2018	NMR		compared to nmrML and NMRSTAR tries to not be exhaustive but limit itself to core set of parameters, carefully include structure assignment. Maintained by the NMReDATA initiative	SDF
nmrML ⁴⁸⁴	2018	NMR		one goal was to follow the design of mzML, the motivation was to address the different “dialects” of JCAMP-DX implementations	XML
GEMD	2019	links materials, the processes that produce them, and the measurements that characterize them. It resolves some problems that were present in the data model underlying the PIF		JSON	
ord-schema	2019	organic reactions	reactions	The developers built this schema based on a survey they conducted in late 2019/early 2020 with the clear goal to create datasets that can help organic reaction prediction. The consortium is lead by a Governing Committee with members from industry (Google, Merck, Pfizer) and academia	protocol buffer
XDL	2020	organic reaction protocols	reactions	Developed to compile machine-readable experimental scripts that can be executed by robots such as the “chemputer”. ⁸⁵	XML
AIF ¹⁰⁶	2021	gas de/adsorption isotherms		The authors of the Adsorption Information File provide tools convert proprietary files to AIF	STAR
mzMLb ⁴⁸⁵	2021	mass spectrometry	spec-	Uses HDF5 in an attempt to address performance issues with mzML, preserves the mzML structure. Attempts to improve over mz5 by preserving the link to also future mzML versions	HDF5

Table 6: Some examples (non-comprehensive) of schema that have been developed for chemistry and materials science. A more comprehensive overview can be found on <https://fairsharing.org/> via a search for “standards”. There have been many efforts that did not find widespread use.

schema	description
Analytical Information Markup Language (AnIML) ⁴⁵⁷	focused on analytical chemistry and biological data, developed by an ASTM (American Society for Testing and Materials) working group. Found already support by some manufacturers like Agilent or ELNs like LabWare. BSSN software (now owned by Merck) implemented converters for more than 150 instruments ⁴⁵⁸
autoprotocol ⁴⁵⁹	development focused on defining experimental plans in the life sciences that can then be remotely executed, ⁴⁵⁹ but has elements that are general (e.g., compounds, inventory) and cover different aspects of chemistry. Notably, it has a mechanism for the proposal of changes to the standard (Autoprotocol Standard Changes)
Physical Information File (PIF) ⁴⁶⁰	designed to be able to describe a broad array of data “from parts in a car down to a single monomer in a polymer matrix”, hence can have a nesting of subsystems. Maintained by Citrine Informatics
Graphical Expression of Materials Data	Developed by Citrine Informatics, links materials, the processes that produce them, and the measurements that characterize them. It resolves some problems that were present in the data model underlying the PIF
ThermoML ⁴⁶¹	IUPAC standard for storage and exchange of experimental thermophysical and thermochemical property data. Data produced by the NIST Thermodynamics Research Center are provided in this format
Chemical Markup Language (CML) ^{462,463}	is supported by many common programs and has been extended, for example, for spectra (CMLSpect). ⁴⁶⁴ One core idea of the designers of the CML was that the information should be validatable—for example, to automatically find if there are negative atom counts
schema for the Open Reaction Database (ord-schema)	the developers built this schema based on a survey they conducted in late 2019/early 2020 with the clear goal to create datasets that can help organic reaction prediction. Governing committee around participants from industry (Google, Merck, Pfizer) and academia (MIT, Princeton)
XDL ⁸⁵	Developed to compile machine-readable experimental scripts that can be executed by robots
OECD Harmonized Templates	data formats for information on chemicals and safety information
Materials Schema	effort to extend schema.org (a large effort that creates schema for structured data on the web) which would allow indexing by major search engines. This effort is inspired by the bioschemas initiative that has been developing types for the life sciences to increase the findability of data. Currently being developed (pre-alpha version) by NIST
Allotrope Data Format (ADF)	consortium of pharmaceutical industries and a larger partner network of instrument manufacturers with the aim to create a data infrastructure that captures the full lifetime of a sample in the lab. It has been working on an ontology (that is available under a permissive license), standard data schema/file, and already has good coverage for many analytical techniques. Many developments (like access to the source code) require membership in the consortium
Unified Data Model (UDM)	format to store reaction data (schema, conditions, provenance, etc.) and biological testing. Has been maintained by Elsevier, who transferred the ownership to the Pistoia Alliance.

Table 7: Examples of common characterization techniques, file formats, and the conversion tools that have been developed by the cheminfo team (GitHub organizations cheminfo, cheminfo-js, cheminfo-py, mljs, image-js).

technique	file formats	conversion library
pXRD	Bruker brml, PowDLL xy	xrd
adsorption isotherms	Belsorp xls, DVS csv, Micro-metrics csv and txt, IGA txt	isotherm-analysis
thermal gravimetric analysis (TGA)	Perkin Elmer csv and txt, TA Instruments txt	tga-spectrum
x-ray photoelectron spectroscopy (XPS)	VAMAS	vamas
nuclear magnetic resonance spectroscopy (NMR)	Bruker zip, JEOL jdf	brukerconverter, jeolconverter
liquid/gas chromatography (LC/GC)	NetCDF	netcdfjs
mass spectrometry (MS)	mzData, mzML, mzXML	mzData
images (transmission/scanning electron microscopy)	TIFF, png	tiff, fast-png

Table 8: Some initiatives that work on the standardization, digitization, and reusability of chemical data.

initiative	description
GOFair Chemistry Implementation Network	one of the ambitious goals is to create a management structure for standards through IUPAC, which has several ongoing projects concerning the developments of data standards ⁴⁷⁰
Research Data Alliance Chemistry Research Data interest group	aims to establish standards and ontologies and also proposes to involve instrument manufacturers in the discussions
RDA/CODATA Materials Data, Infrastructure & Interoperability interest group	aims to foster the exchange of computational and experimental materials data via interoperability and shared standards
Materials Research Data Alliance	emerged from NSF's 2019 Summit on Big Data and Materials Cyberinfrastructure with the goal of creating a community around materials data sharing and reuse
National Research Data Infrastructure for Chemistry (NFDI4Chem)	German consortium with the aim to build an open and FAIR infrastructure for research data management in chemistry

B

SUPPORTING INFORMATION
FOR "AN ECOSYSTEM FOR
DIGITAL RETICULAR
CHEMISTRY"B.1 GROWTH OF MACHINE LEARNING FOR RETIC-
ULAR CHEMISTRY AND POROUS MATERI-
ALS

To illustrate the importance of machine learning for reticular chemistry and porous materials, we conducted a literature survey using Scopus, interfaced via pybibliometrics.⁴⁸⁶ We used TITLE-ABS-KEY(("reticular chemistry" OR "metal-organic framework" OR "covalent-organic framework" or "zeolite") AND ("machine learning" OR "neural network" OR "data-driven" OR "deep learning" OR "random forest" OR "gradient boost" OR "support vector" OR "regression" OR "recommendation system" OR "inverse design" OR "recommender system" OR "active learning" OR "bayesian optimization")) as query and plot in Figure 47 the number of publications aggregated per year (excluding 2023, which already counted 7 matches at the time we last performed this survey, December 12, 2022).

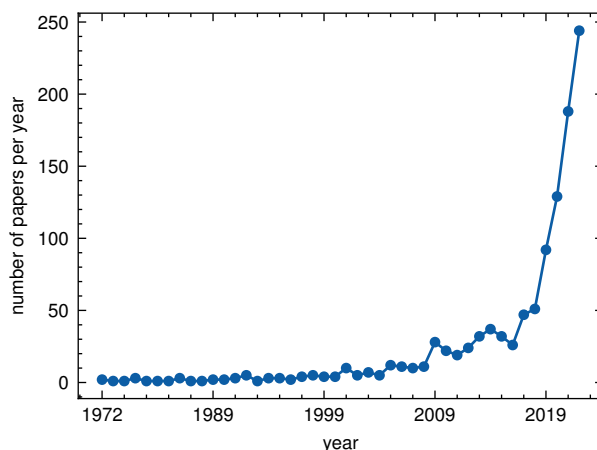


Figure 47: Count of publications aggregated per year that mention some concept of reticular chemistry or porous materials besides machine learning concepts.

B.2 GAS ADSORPTION REFERENCE DATASET

Note that this dataset contains a significant fraction of zeros for cases in which the simulation was skipped because they were deemed inaccessible to the guest following geometric analysis. This is because the QMOF dataset focused, for reasons of

computational cost, on the subset of MOFs with small unit cells. Additionally, it is important to realize that the MOF subset of the CSD¹¹⁶ contains a large fraction of nonporous materials (best described as coordination polymers).

B.2.1 Computed properties

We used the DDEC6 charges^{487–490} from the PBE-D3(BJ) calculations as provided with the QMOF database. We always described the framework using the UFF²¹⁷ forcefield, applying analytical tail-corrections²¹⁸ for the contributions after the cut-off of 12 Å. For zeo++,¹⁷⁶ we always used 100,000 samples for the calculation of the probe-occupiable volume and 100 Å^{−3} samples for the computation of the blocked pockets.

CO₂ and N₂ isotherms were sampled using the algorithm described in Ongari et al.¹⁴⁷ For other isotherms, we used a fixed grid of pressure points. We always employed 100,000 cycles with the RASPA code for the computation of the Henry coefficients.²¹⁶ The additional simulation-specific settings are detailed in the following Tables. The full provenance graph can be downloaded on the MaterialsCloud (10.24435/materialscloud:qt-cj).

Table 10: Simulation parameters for CO₂ isotherms (and Widom insertions).

<i>parameter</i>	<i>value</i>
force field guest	TraPPE ⁴⁹¹
saturation density / mol L ^{−1}	21.2
probe radius / Å	1.525
<i>T</i> / K	300
initialization cycles	1000
production cycles	10000
<i>p</i> sampling precision	0.1
max distance between <i>p</i> points / bar	5
lowest pressure point / bar	0.001
largest pressure point / bar	30

Table 11: Simulation parameters for N₂ isotherms (and Widom insertions).

<i>parameter</i>	<i>value</i>
force field guest	TraPPE ⁴⁹¹
saturation density / mol L ^{−1}	28.3
probe radius / Å	1.655
<i>T</i> / K	300
initialization cycles	1000
production cycles	10000
<i>p</i> sampling precision	0.1
max distance between <i>p</i> points / bar	5
lowest pressure point / bar	0.001
largest pressure point / bar	30

For the process simulations, we used very simplified models of a temperature–

Table 12: Simulation parameters for H₂ isotherms (and Widom insertions).

<i>parameter</i>	<i>value</i>
force field guest	MDT+DL (as described in Bucior et al. ²³⁰ , i.e., dispersion from Michels et al. ⁴⁹² and charges from Darkrim and Levesque ⁴⁹³)
saturation density / mol L ⁻¹	35.4
probe radius / Å	1.48
initialization cycles	3000
production cycles	3000
temperature grid / K	77, 198, 298
pressure grid / bar	1.0, 5.0, 25, 50, 75, 100

Table 13: Simulation parameters for CH₄ isotherms (and Widom insertions).

<i>parameter</i>	<i>value</i>
force field guest	TraPPE ⁴⁹⁴
saturation density / mol L ⁻¹	26.34
probe radius / Å	1.865
<i>T</i> / K	298
initialization cycles	1000
production cycles	10000
pressure grid / bar	1.0, 5.8, 20, 35, 50, 65

pressure swing process¹⁴⁸ in which we assume a constant heat capacity for all materials, which is known not to be a correct approximation.¹³²

B.2.2 Dataset description

The following Figures show the distributions of the computed properties.

B.3 DUPLICATES

Table 14: Simulation parameters for O₂ isotherms (and Widom insertions). Based on settings chosen in Moghadam et al.⁴⁹⁵

<i>parameter</i>	<i>value</i>
force field guest	TraPPE ⁴⁹⁶
saturation density / mol L ⁻¹	71.3
probe radius / Å	1.51
<i>T</i> / K	298
initialization cycles	5000
production cycles	5000
pressure grid / bar	1, 5, 10, 20, 30, 50, 80, 100, 140, 200

Table 15: Simulation parameters for Xe Widom insertions.

<i>parameter</i>	<i>value</i>
force field guest	BOATO ⁴⁹⁷
saturation density / mol L ⁻¹	22.4
probe radius / Å	1.985

Table 16: Simulation parameters for Kr Widom insertions.

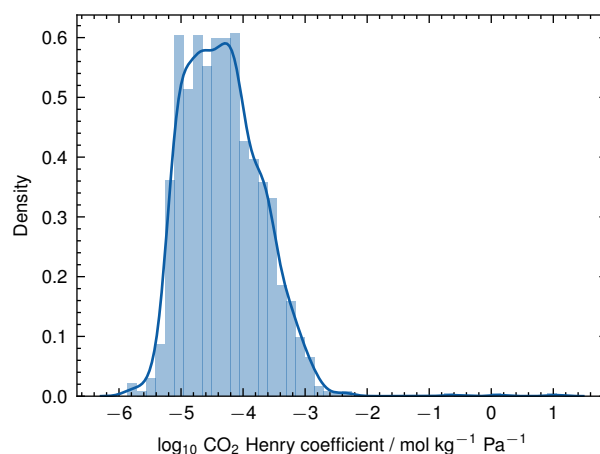
<i>parameter</i>	<i>value</i>
force field guest	BOATO ⁴⁹⁷
saturation density / mol L ⁻¹	29.0
probe radius / Å	1.83

Table 17: Simulation parameters for water Widom insertions.

<i>parameter</i>	<i>value</i>
force field guest	TIP4P/2005 ⁴⁹⁸
saturation density / mol L ⁻¹	53.3
probe radius / Å	1.58

Table 18: Simulation parameters for H₂S Widom insertions.

<i>parameter</i>	<i>value</i>
force field guest	ESP-MM ⁴⁹⁹
saturation density / mol L ⁻¹	26.9
probe radius / Å	1.74

Figure 48: Distribution of \log_{10} CO₂ Henry coefficient / mol kg⁻¹ Pa⁻¹.

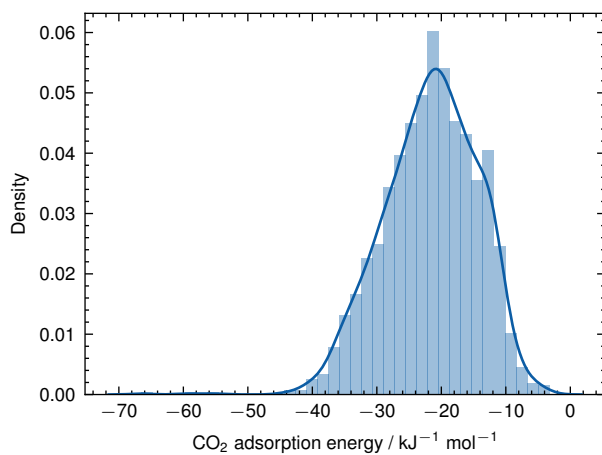


Figure 49: Distribution of \log_{10} CO₂ adsorption energy / kJ mol⁻¹.

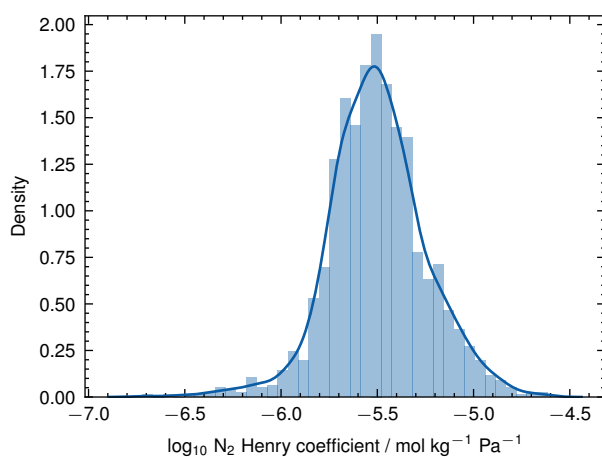


Figure 50: Distribution of \log_{10} N₂ Henry coefficient / mol kg⁻¹ Pa⁻¹.

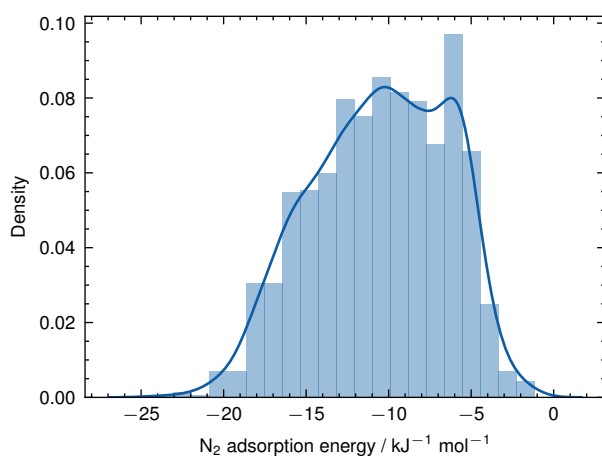


Figure 51: Distribution of \log_{10} N₂ adsorption energy / kJ mol⁻¹.

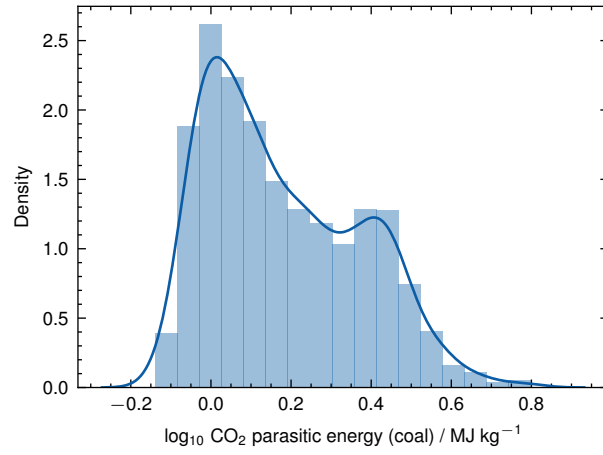


Figure 52: Distribution of $\log_{10} \text{CO}_2$ parasitic energy (coal) / MJ kg^{-1} .

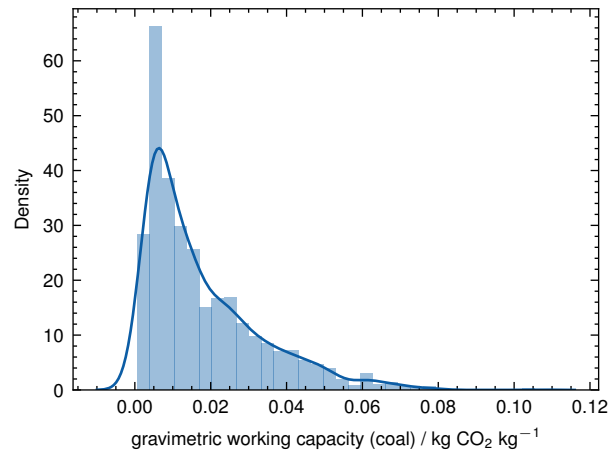


Figure 53: Distribution of $\log_{10} \text{CO}_2$ gravimetric working capacity (coal) / $\text{kg CO}_2 \text{ kg}^{-1}$.

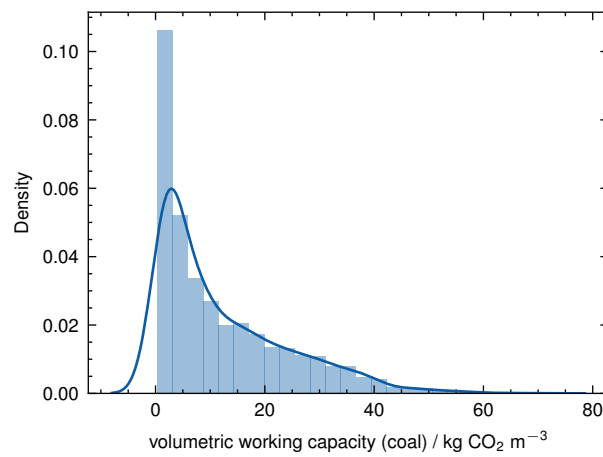


Figure 54: Distribution of \log_{10} volumetric working capacity (coal) / $\text{kg CO}_2 \text{ m}^{-3}$.

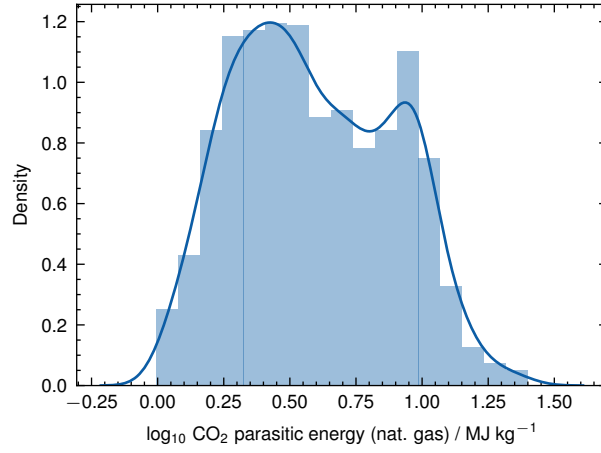


Figure 55: Distribution of $\log_{10} \text{CO}_2$ parasitic energy (nat. gas) / MJ kg^{-1} .

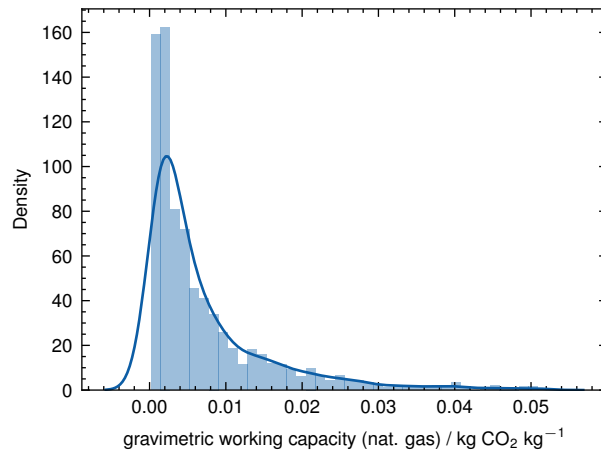


Figure 56: Distribution of $\log_{10} \text{CO}_2$ gravimetric working capacity (nat. gas) / $\text{kg CO}_2 \text{ kg}^{-1}$.

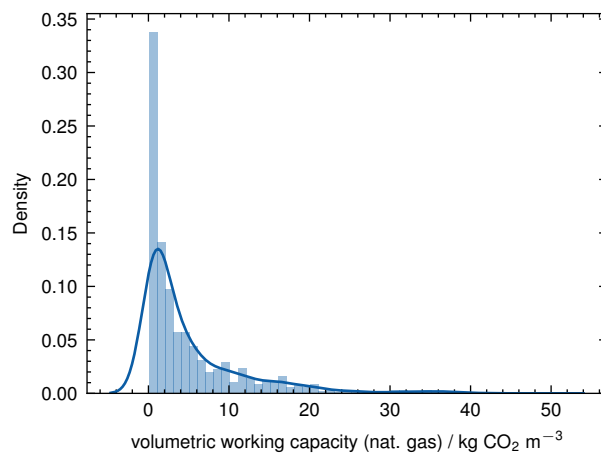


Figure 57: Distribution of \log_{10} volumetric working capacity (nat. gas) / $\text{kg CO}_2 \text{ m}^3$.

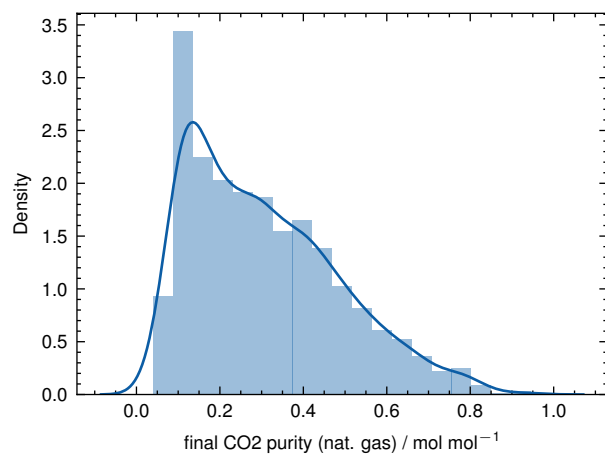


Figure 58: Distribution of \log_{10} final CO₂ purity (nat. gas) / mol mol⁻¹.

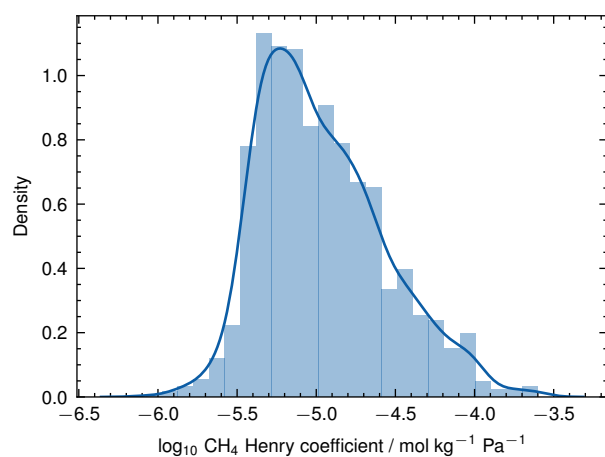


Figure 59: Distribution of \log_{10} CH₄ Henry coefficient / mol kg⁻¹ Pa⁻¹.

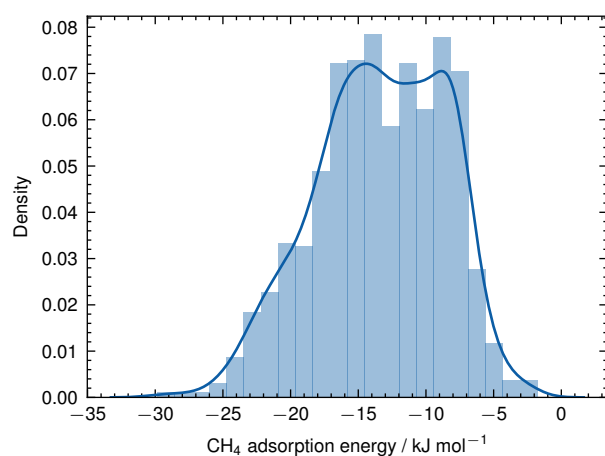


Figure 60: Distribution of \log_{10} CH₄ adsorption energy / kJ mol⁻¹.

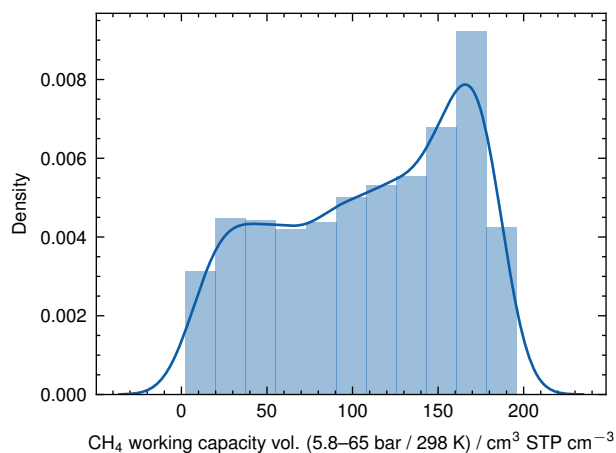


Figure 61: Distribution of \log_{10} CH₄ working capacity vol. (5.8-65 bar/298 K) / cm³_{STP} cm⁻³.

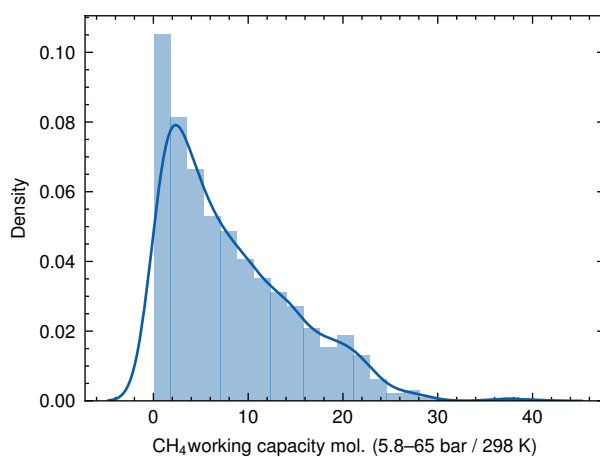


Figure 62: Distribution of \log_{10} CH₄ working capacity (5.8-65bar/298K) / mol kg⁻¹.

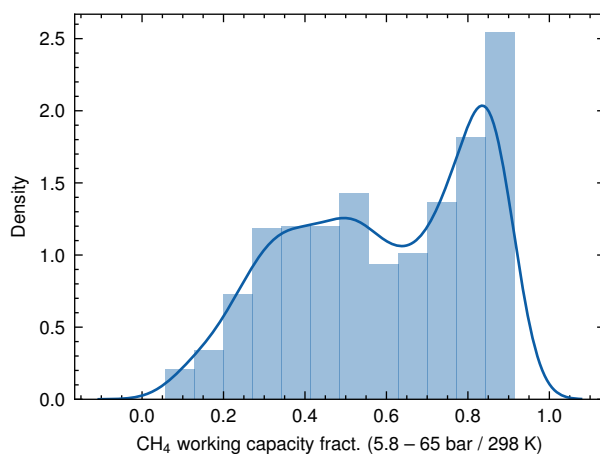


Figure 63: Distribution of \log_{10} CH₄ working capacity fract. (5.8-65 bar/298 K).

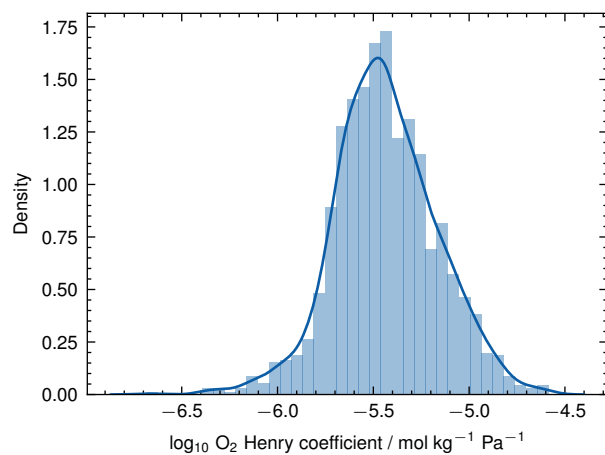


Figure 64: Distribution of $\log_{10} \text{O}_2$ Henry coefficient / $\text{mol kg}^{-1} \text{Pa}^{-1}$.

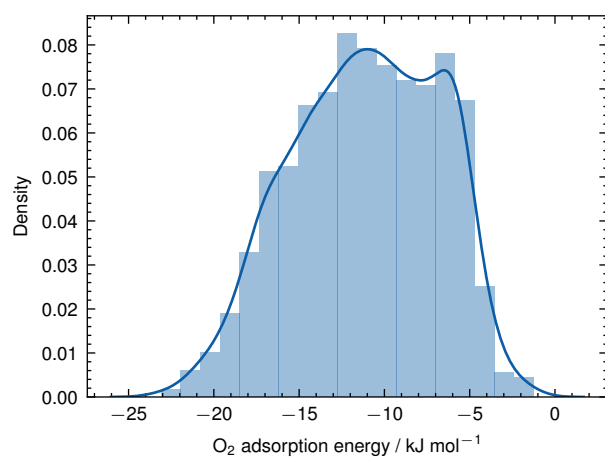


Figure 65: Distribution of $\log_{10} \text{O}_2$ adsorption energy / kJ mol^{-1} .

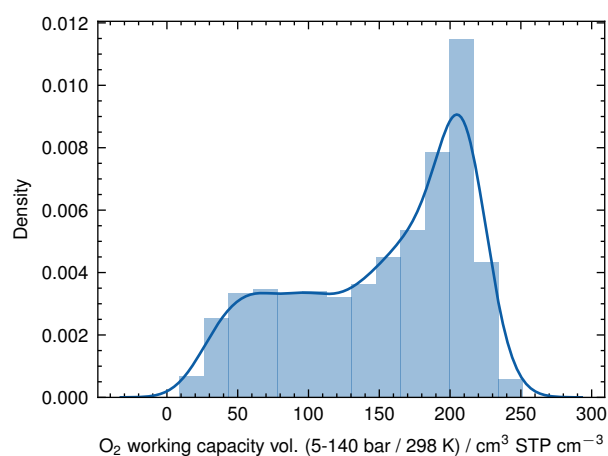


Figure 66: Distribution of $\log_{10} \text{O}_2$ working capacity (5-140 bar/298 K) / $\text{cm}^3_{\text{STP}} \text{cm}^{-3}$.

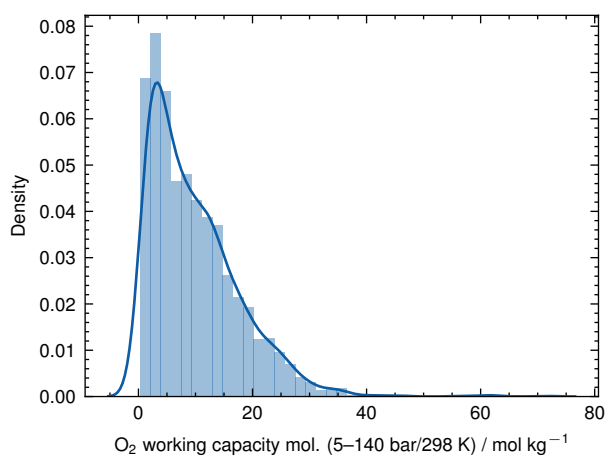


Figure 67: Distribution of \log_{10} O₂ working capacity (5-140 bar/298 K) / mol kg⁻¹.

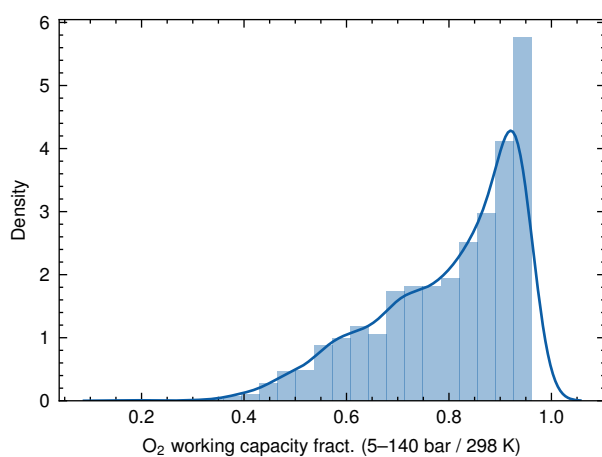


Figure 68: Distribution of \log_{10} O₂ working capacity (5-140 bar/298 K).

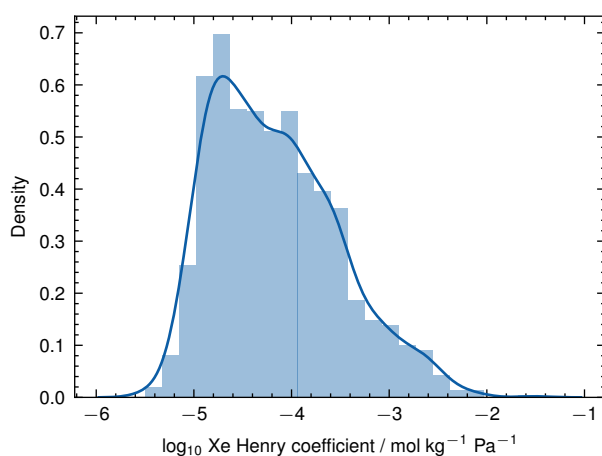


Figure 69: Distribution of \log_{10} Xe Henry coefficient / mol kg⁻¹ Pa⁻¹.

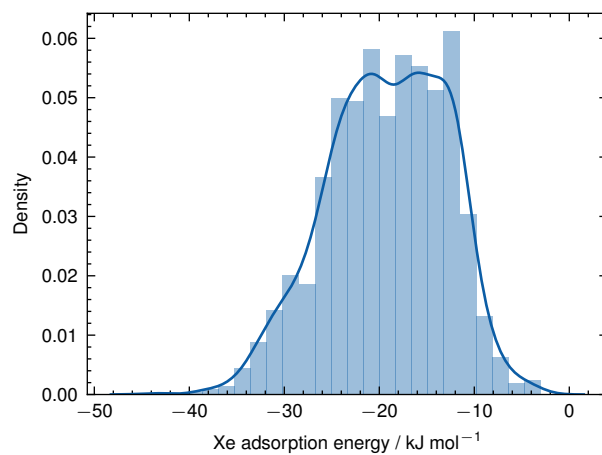


Figure 70: Distribution of \log_{10} Xe adsorption energy / kJ mol^{-1} .

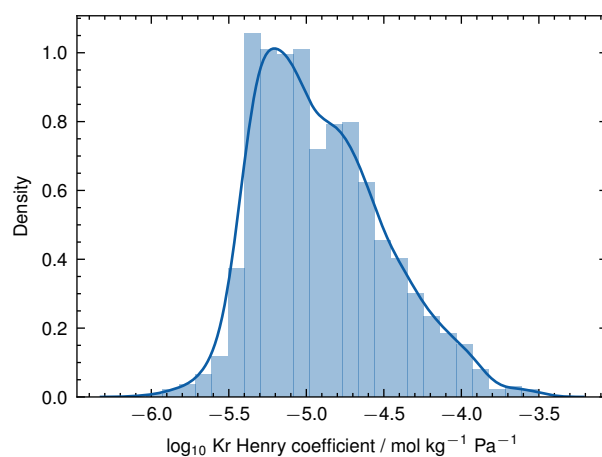


Figure 71: Distribution of \log_{10} Kr Henry coefficient / $\text{mol kg}^{-1} \text{Pa}^{-1}$.

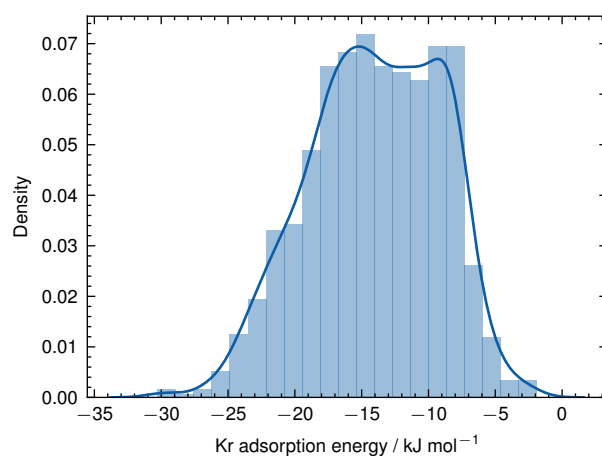


Figure 72: Distribution of \log_{10} Kr adsorption energy / kJ mol^{-1} .

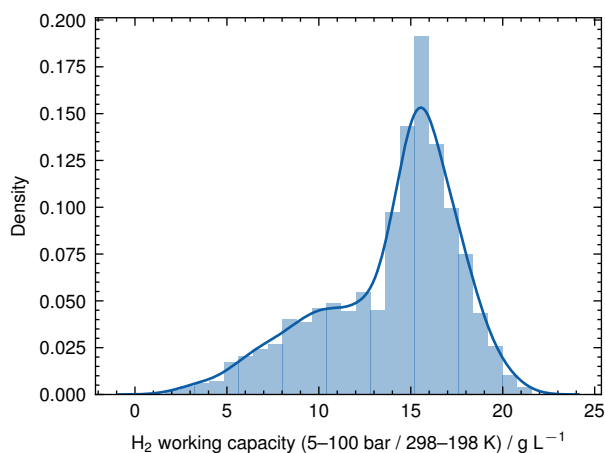


Figure 73: Distribution of \log_{10} H₂ working capacity (5-100 bar/298-198 K) / g L⁻¹.

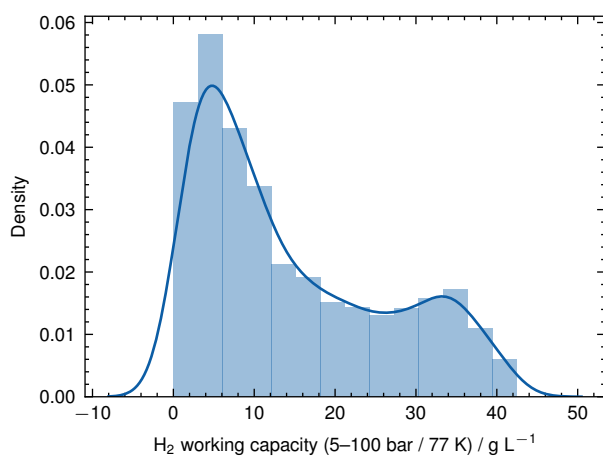


Figure 74: Distribution of \log_{10} H₂ working capacity (5-100 bar/77 K) / g L⁻¹.

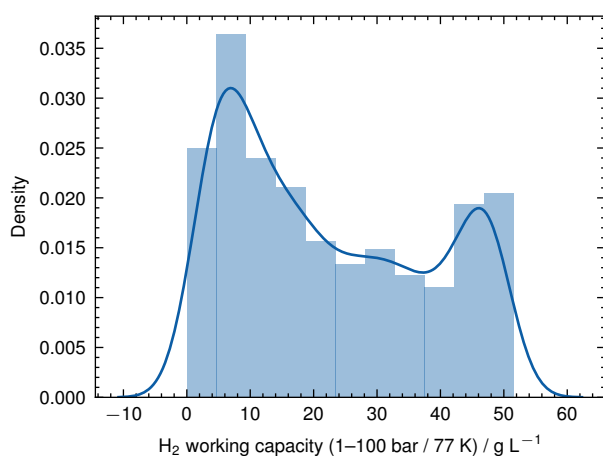


Figure 75: Distribution of \log_{10} H₂ working capacity (1-100 bar/77 K) / g L⁻¹.

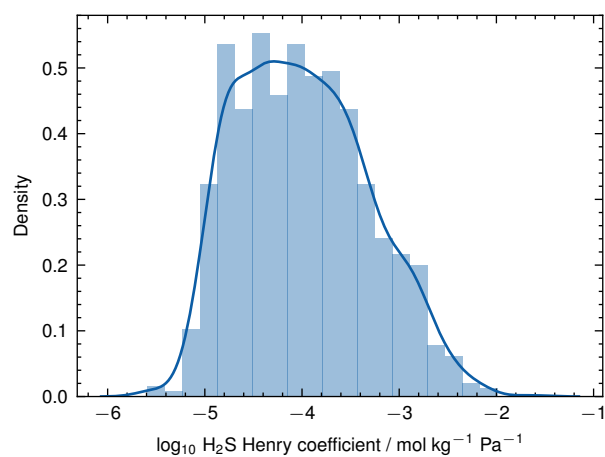


Figure 76: Distribution of $\log_{10} \text{H}_2\text{S}$ Henry coefficient / $\text{mol kg}^{-1} \text{Pa}^{-1}$.

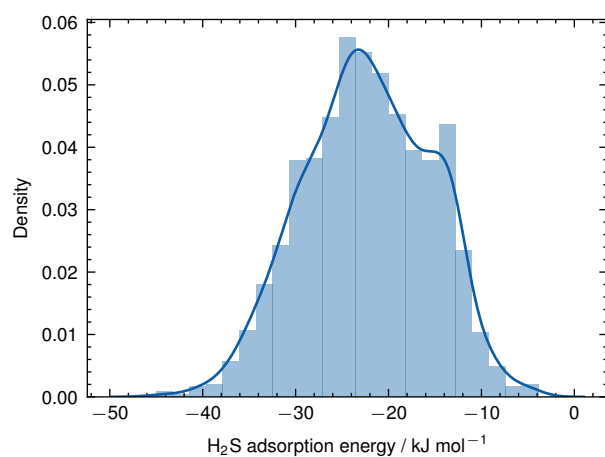


Figure 77: Distribution of $\log_{10} \text{H}_2\text{S}$ adsorption energy / kJ mol^{-1} .

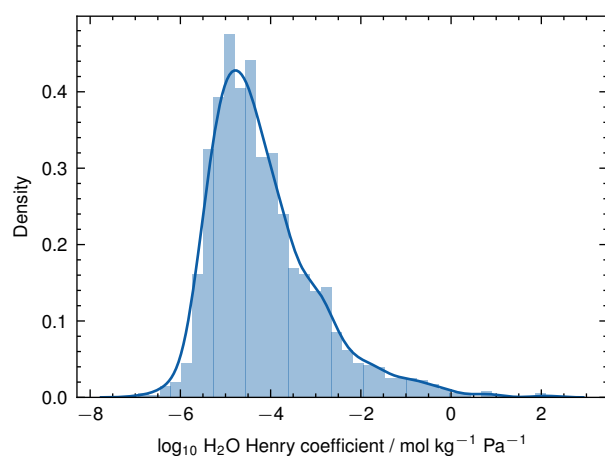


Figure 78: Distribution of $\log_{10} \text{H}_2\text{O}$ Henry coefficient / $\text{mol kg}^{-1} \text{Pa}^{-1}$.

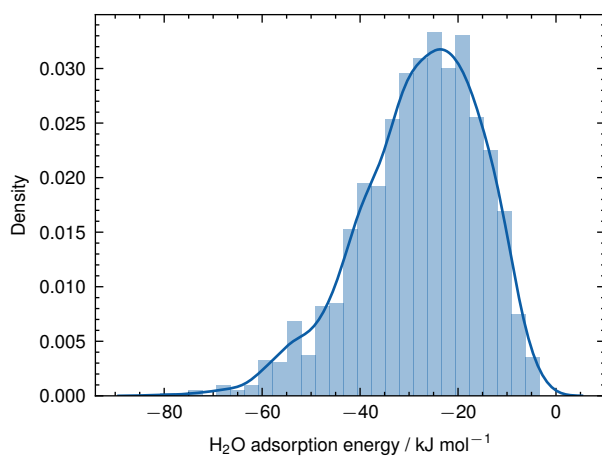


Figure 79: Distribution of \log_{10} H_2O adsorption energy / kJ mol^{-1} .

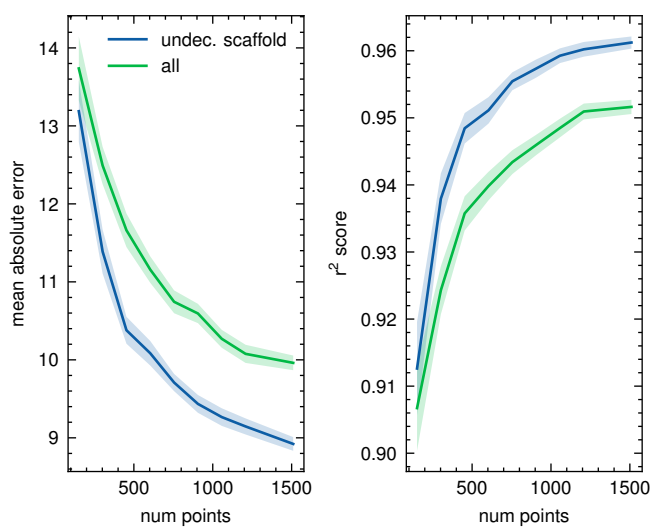


Figure 80: Learning curves with all data points and only unique undecorated scaffolds (BW dataset and CH_4 deliverable capacity as the target, using XGBoost regressor on the default dataset). The shaded area indicates 95 % standard intervals. For this case study, dropping undecorated scaffold duplicates leads to improved performance metrics.

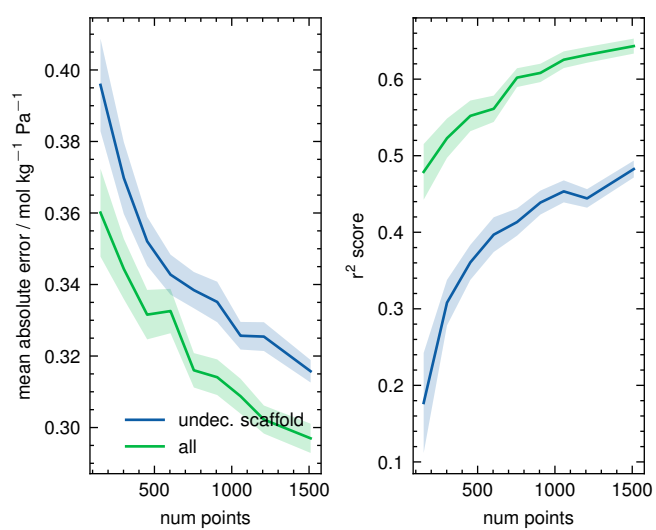


Figure 81: Learning curves with all data points and only unique undecorated scaffolds (CoRE dataset and CO₂ Henry coefficient as the target, using XGBoost regressor on the default dataset). The shaded area indicates 95 % standard intervals.

B.4 FEATURIZERS

B.4.1 Addition of aggregations

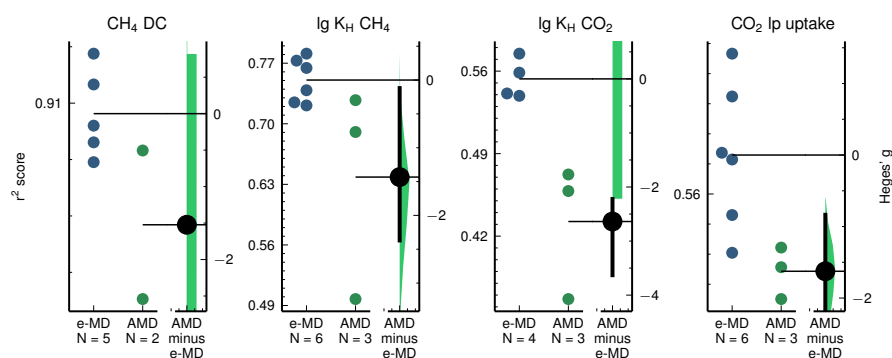


Figure 82: Minimum distance fingerprint with and without additional aggregations. The original average minimum distance fingerprints as proposed by Widdowson et al. aggregate point-wise distance distributions (PDD) using (weighted) averages (top row). In mofdscribe, users can compute other aggregations such as min, max, and the standard deviation (std, bottom row, e-MD). To ensure a fair comparison, we optimize the *full* pipeline (including pre-processing and the model) using automated machine learning.¹⁸⁵ The plots visualize the measured model performance and estimated effect sizes in terms of Hedges' g.⁵⁰⁰ The blue points always indicate the coefficient of determination (r² on a holdout test set, measured on the left axes) of the models trained with additional aggregations (e-MD.) The green ones indicate the coefficients of determination of the models trained with only the mean (AMD) as aggregation. To quantify the effect, we bootstrap the Hedges' g (a suitable effect size metric in the case of little data,¹⁷⁰ shown on the right axes) and show it with a kernel density estimate. In all cases, the addition of chemistry shows very large effects.¹⁸⁶

B.5 GRAPH HASHES

Weisfeiler and Lehman devised a graph-isomorphism test based on iterative refinement of graph colorings to derive a canonical form.¹⁵⁵ Two non-isomorphic graphs might share the same canonical form; however, if the canonical forms are not identical, the graphs are definitely not isomorphic. Therefore, this test might lead us to identify too many isomorphic graphs. However, since our objective is mostly duplicate removal to avoid data leakage, this is preferable to missing duplicates.

In practice, we encode the periodic crystal graph as a labeled quotient graph (LQG) (labels indicating into which periodic image the bonding extends) but do not consider the directions and edge voltages for the hash derivation. We rely on the fact that two LQGs of the same crystallographic net cannot have non-isomorphic unlabeled quotient graphs (UQGs). Hence, computing a hash of the Weisfeiler-Lehman canonical form of the UQG will always lead to too many duplicates, not too few.

B.6 SPLITTERS

Our main objectives for the default settings in the splitter classes are:

- To minimize data leakage. To ensure this, we implement grouped splits. Typically, we group on undecorated scaffold hashes.
- To minimize imbalance effects on the imbalance. To ensure this, we implement stratification. Typically, we stratify on the target (and bin if the target is continuous*).

B.6.1 Grouped and stratified holdout splits

While we can rely on `sklearn`'s implementations for the grouped k -fold cross-validation case, there is no off-the-shelf implementation for the grouped and stratified case (for k -fold cross-validation and partitioning). Therefore, we implement the following algorithm in which we perform the following steps:

- Assign each structure to a group, for instance, based on the undecorated scaffold hashes.
- Aggregate the target property within each group, e.g., using the arithmetic mean.
- Use the aggregated properties for a stratified split
- For each group, add all the members

While this algorithm does not guarantee that we will match the requested train/test ratio, it guarantees grouping and some level of stratification, which are more important objectives for a stringent model evaluation.

B.6.2 Case studies

HYPERPARAMETER GRID

- `colsample_bylevel`: on logarithmic grid from 0.01 to 0.1
- `depth`: integers between 1 and 16
- `iterations` integer between 1 and 10000
- `learning_rate` float on log scale between 0.001 and 0.5
- `l2_leaf_reg`: float between 0.01 and 10
- `random_strength`: float between 0.01 and 10
- `bagging_temperature`: float between 0.01 and 10

B.7 MOF FRAGMENTATION

The MOF fragmentation algorithm is outlined in Algorithm 1. Therein, we use the following definitions:

- *branching site* is a site that fulfills the following conditions:
 - has at minimum coordination number 3

* Note that this is already a weak source of data leakage as we need to consider the full dataset for binning. Also, note that there is still debate on the pros and cons of this approach, see <https://github.com/kjappelbaum/mofdscribe/discussions/242>.

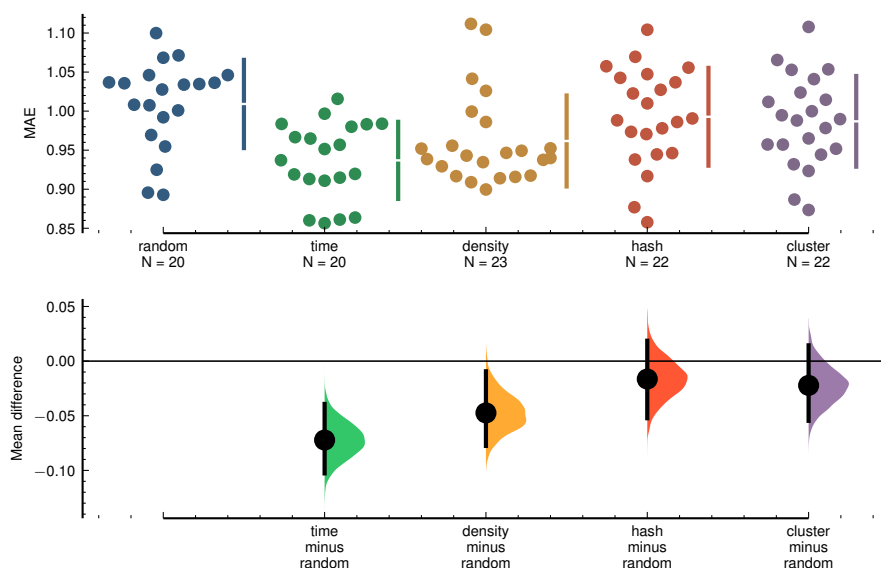


Figure 83: Bootstrapped mean difference in mean absolute error for predicting low-pressure CO₂ uptake. The model was trained on the CoRE dataset in `moFdescribe` with the default feature set and tested on the ARABG dataset.

- has at least one path with maximum 2 edges that leads to metal and does not contain a bride
- has at minimum 2 non-metal connections that are not bridges

If there are multiple neighboring sites selected according to this definition, we pick the one closest to the metal (the fewest number of edges).

- *bridge* is an edge that, when broken, increases the number of connected components
- *connected component* is a connected subgraph that is *not* part of any larger connected subgraph

Data: MOF

Result: fragments and net

```
unbound_solvent = locate_unbound_solvent(MOF)
```

```
ignored_metals = []
```

```
while potential_metal_in_linker do
```

```
    node_candidates = locate_nodes(MOF, unbound_solvent,
                                   ignored_metals)
```

```
    bound_solvent = locate_bound_solvent(MOF, node_candidates)
```

```
    linker_candidates = locate_linkers(MOF, node_candidates,
                                       bound_solvent)
```

```
    metals_to_ignore, potential_metal_in_linker =
        check_metal_in_linker(MOF, linker_candidates)
```

```
end
```

```
net = build_net(MOF, node_candidates, linker_candidates)
```

Algorithm 1: Fragmentation pseudocode. The `while` loop ensures that we do not classify metal-containing linkers (e.g., porphyrins) as metal clusters. Typically, only one—or in the case of porphyrin linkers—two loops are performed.

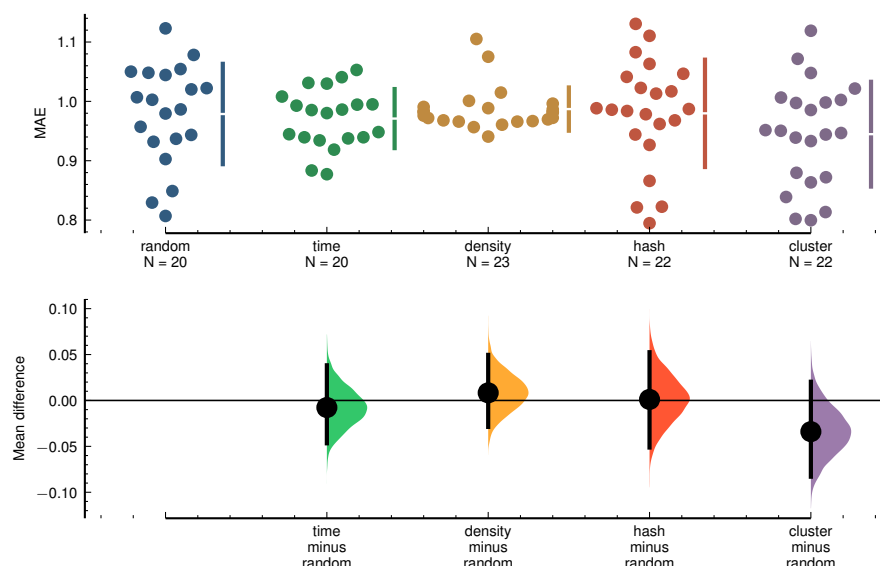


Figure 84: Bootstrapped mean difference in mean absolute error for predicting low-pressure CO₂ uptake. The model was trained on the CoRE dataset in `mofdscribe` with the default feature set and tested on the BW dataset.

THE `locate_unbound_solvent` FUNCTION creates a $3 \times 3 \times 3$ supercell and analyzes if there are any non-periodic connected components. Those are floating molecules in the cell. In practice, we use a customized version of the `get_subgraphs_as_molecules` function in `pymatgen`.¹⁷⁴

THE `locate_nodes` FUNCTION performs depth-first-search between all metals (ignoring those on the `ignored_metals` list) and potential branching sites. After clustering of neighboring branching sites and simplification in case multiple neighboring branching sites are found, we identify the connected components that are spanned by the depth-first-search paths between metals and branching sites (where needed, we complete this path by also including, for example, bound hydrogen atoms on oxygen atoms connecting metals and branching sites). Those connected components are the metal nodes.

THE `locate_bound_solvent` FUNCTION checks for bridges on the node candidates. Note that, by default, we do not break those bridges. That is, the bound solvent remains bound to the nodes.

THE `locate_linkers` FUNCTION identifies the remaining connected components. For this, it deletes all solvent and node vertices (and the associated edges) from the structure graph. However, it keeps the branching indices. The remaining connected components are the linkers.

THE `check_metal_in_linker` FUNCTION attempts to identify linkers containing metals (e.g. porphyrins) that were incorrectly identified as metal clusters. For this, it checks for the co-planarity of the metal and the branching sites.

We also ensure that the output molecules (in the linker and metal cluster collections) are correctly unwrapped by performing a breadth-first-search over the structure graph and picking the Cartesian coordinates of the neighbor image closest to the current Cartesian coordinates.

```

# load a CIF
mof = MOF.from_cif('tests/test_files/HKUST-1.cif')

# Fragment the MOF
fragments = mof.fragment()

# If you are in a Jupyter notebook you can visualize the components.
fragments.linkers[0].show_molecule()
fragments.nodes[0].show_molecule()

# You can also search PubChem for the building blocks
fragments.linkers[0].search_pubchem()

# To get the [RCSR code](http://rcsr.anu.edu.au/nets) run
fragments.net_embedding.rcsr_code

```

Listing 5: Example for the use of the `moffragmentor`.

THE build_net FUNCTION uses the Cartesian coordinates of branching sites on metal clusters on linkers to identify connected building blocks and their barycenters. Additionally, we remove 2-connected vertices. To obtain an RCSR code, we input the labeled quotient graph in CGD format to the Systre program.⁵⁰¹

All building blocks (metal clusters, linkers, solvent molecules) are stored in dedicated Python objects with wrapped molecules, original coordinates, and branching indices (among others) as attributes.

An example of the use of `moffragmentor` is given in Listing 5

B.8 LEADERBOARD

Our website hosts multiple task-specific leaderboards. Each leaderboard has an interactive plot (comparing the metrics) along with a data table. The plots and data tables are automatically populated based on the `json` output of a `bench` run. In particular, we use custom Sphinx directives (via the `sphinxcontribs-needs` package) to standardize the metrics and implement interactive data tables with filtering functionality.

The pull request template we ask users to fill upon contributing a new model not only asks for the `json` file summarizing the metrics but also for a restructured text (`.rst`) file describing the model. This template also contains several questions inspired by the model cards proposed by Kapoor and Narayanan¹³⁰, intended to encourage submitters to reflect on potential data leakage.

C

SUPPORTING INFORMATION FOR “BIAS FREE MULTIOBJECTIVE ACTIVE LEARNING FOR MATERIALS DESIGN AND DISCOVERY”

C.1 NOTES ON ACTIVE LEARNING AND PARETO DOMINANCE

c.1.1 Problem setting considered in this work

In this work, we are interested in *classifying* with confidence a set of observations in objective space as (approximate)-Pareto optimal points or as non-dominating points, with a particular focus on sampling from regions of design space near the Pareto optimal points.

c.1.2 Order theory

A partial order is a binary relation* \succeq on a set Ω that satisfies

1. reflexivity: $x \succeq x$
2. antisymmetry: $x^1 \succeq x^2$ and $x^2 \succeq x^1$ imply $x = y$
3. transitivity: $x^1 \succeq x^2$ and $x^2 \succeq x^3$ imply $x^1 \succeq x^3$

for all $x^1, x^2, x^3 \in \Omega$.

The Pareto dominance relationship defines a partial order that is additionally scale and translation invariant as follows:

1. translational invariance: $\forall x \in \mathbb{R}^m : x^1 \preceq x^2 \Rightarrow x^1 + x \preceq x^2 + x$
2. scale invariance: $\forall \alpha \in \mathbb{R}^+ : x^1 \preceq x^2 \Rightarrow \alpha x^1 \preceq \alpha x^2$

In contrast, a total order is a partial order that fulfills the comparability axiom:

$$\forall x^1, x^2 \in \Omega : x^1 \preceq x^2 \vee x^2 \preceq x^1$$

A set with a total order defines a chain. To induce a total order, one hence needs to introduce a bias.

Clearly, we want to remove any form of bias in (computational) materials discovery. In the initial design and discovery stage, we are interested in identifying a set of possible candidates to enter the next design stage. That is, we first want to identify the partially ordered set (poset) of optimal candidates, ideally with confidence, without weighting our different objectives. In principle, this can also provide additional insights into whether certain objectives must be weighted differently. It is

* A binary relation, R between sets X and Y , XRY , is a subset of the Cartesian product $X \times Y$

important to realize that for the extreme case when the number of iterations equals the number of design points, we will also be able to identify the unbiased partial order using any approach. However, in typical applications, enumerating through all iterations is unfeasible, and thus we want to use an active learning approach to keep the number of iterations as minimal as possible.

In the following sections, we illustrate how the biases in popular acquisition functions, such as expected improvement, can affect the resulting optimization outcome. For more detail, we refer the reader to Zitzler et al.²⁵², Wagner et al.²⁷¹, del Rosario et al.⁵⁰², and Moffaert and Nowé⁵⁰³ (in the context of reinforcement learning).

c.1.3 Improvement measures

In the single-objective case, it is clear how to measure improvement; it is directly given by the ranking of scalar objective values. In the multi-dimensional case, this improvement measure is no longer well-defined. Commonly used improvement measures introduce a total order in the search space and hence bias the search.

Expected Improvement

There has been an effort to generalize expected improvement (EI) measures that are probabilistically optimal under some assumptions (see below) for multiobjective optimization. In general, expected improvement measures take the form of an integral over the product of the improvement and the probability of improvement over the non-dominated area A , which is represented by factorized normal distributions.

$$EI = \int_{\mathbf{y} \in A} \underbrace{I(\mathbf{y}, \mathbf{P})}_{\text{improvement}} \underbrace{\prod_{i=1}^m \frac{1}{\hat{s}_i(\mathbf{x})} \phi\left(\frac{y_i(\mathbf{x}) - \hat{y}_i(\mathbf{x})}{\hat{s}_i(\mathbf{x})}\right)}_{\text{probability of improvement}} d\mathbf{y}_i(\mathbf{x}) \quad (3)$$

c.1.4 Biases with improvement measures

Only defined on a subset of the Pareto set

The simplest approach to deal with a multiobjective problem is linear scalarization, i.e., mapping the multiobjective problem into a single-objective optimization problem. Here, we develop a weighted sum of our objectives and use this as our overall objective function:

$$L = \sum_i^m w_i y_i. \quad (4)$$

This defines a convex function. Therefore, this approach will fail when parts of a Pareto front are non-convex. One can imagine performing multiple searches with different weighting functions, but this can be burdensome, and it is unclear how to define such weighting functions.⁵⁰⁴

Dimensional inhomogeneity

The first (trivial) bias is dimensional inhomogeneity. This can be nicely exemplified by choosing the improvement measure as the Euclidean distance from a Pareto optimal point to the closet point in the non-dominating set as proposed by Keane⁵⁰⁵ and also implemented by Janet et al.:²⁴⁵

$$I(\mathbf{y}, \mathbf{P}) = \min_{j=1}^k \sqrt{\sum_{i=1}^m (y_i(\mathbf{x}) - y_i^j)^2}, \quad (5)$$

where k iterates over the set of non-dominating points with the current Pareto front \mathbf{P} . Here, we can see that the summation can include distance metrics between observables in different dimensions. The Euclidean norm shares sensitivity to various scaling of the different objectives for all $L_{p>0}$ metrics (i.e., metrics based on a norm $\sum_i \|x_i\|^p$). This can be problematic, as we can imagine that one objective (e.g., $i = 1$) might be on the order 10^{-3} , whereas another (e.g., $i = 2$) might be on the order of 10^6 . Without re-scaling, the second objective would be given a much higher weight. Hence, unless we have *a priori* knowledge of the range of values for different objectives, such re-scaling can be nontrivial. Wagner et al.²⁷¹ have shown that such a metric does not preserve the Pareto dominance relation. In Figure 85, we illustrate additional examples of how EI can be sensitive to such re-scaling methods.

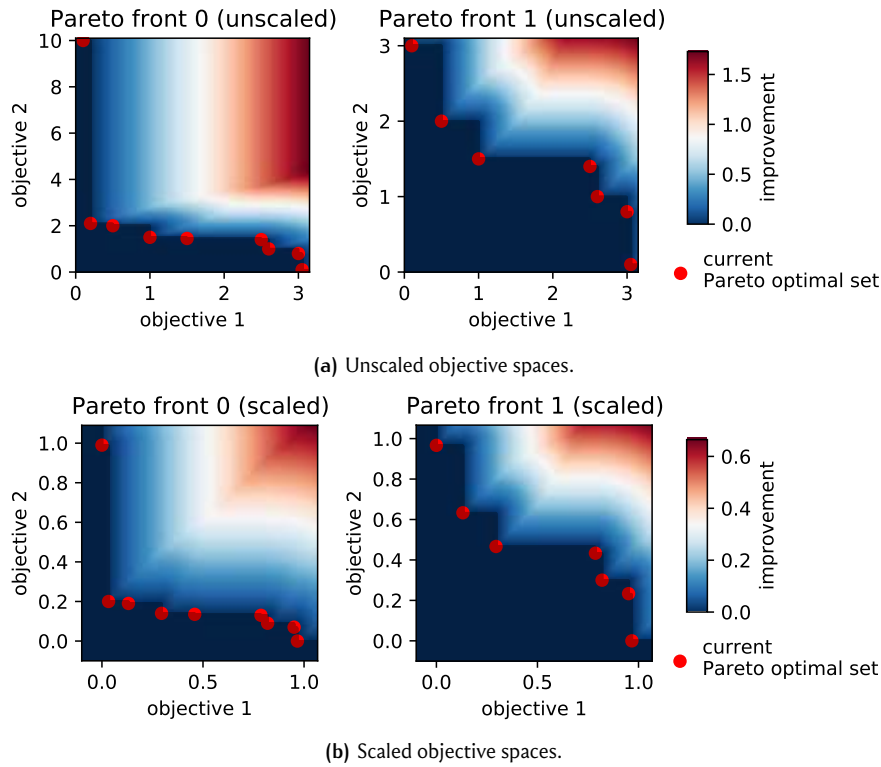


Figure 85: Improvement landscape for two fictitious Pareto frontiers using eq. 5. The color coding indicates the value of the improvement function in each grid point. In a Bayesian optimization scheme, we would choose the sample with the highest value of the improvement function, assuming that the variance is equal for all points. In a material discovery setting, however, there will not be a material on each grid point. A material with a certain combination of objectives might not exist, in which case, there would be regions of space in the figure where the improvement is unknown. Note that this improvement function landscapes include “rifts” that distort the Pareto dominance relation. Also note that the improvement landscape changes when we change the scales. This is evident by comparing **a** with **b**. For example, we see for Pareto front 0 that in the unscaled space improvements in directions of objective 2 get a higher weight, wherefore the aspect ratio of the red region changes upon scaling of the objectives.

Choice of reference point

The hypervolume indicator is known as the only quantitative indicator that is strictly increasing with respect to Pareto dominance.²⁵² However, it can be sensitive to the choice of a reference point.

Moreover, by construction, the hypervolume indicator gives higher weight to the convex part of the Pareto front.²⁵² As shown in Figure 86, we show how improvement measures are greatest (red regions) in the top right portions of the figure panels. Regions within this space that solely improve one objective can contribute to the hypervolume indicator quite drastically depending on the reference point.⁵⁰⁶

c.1.5 Optimization vs. active learning

Expected improvement measures are probabilistically optimal under the assumption that the current sample is in the evaluated set and that the current evaluation is the final evaluation.²⁵¹ In our problem setting, however, we do not assume that the current evaluation is the final evaluation and that this evaluation will be part of the output set.^{251,507} For our active learning approach, we are interested in expediting the classification of points in objective space as (approximate) Pareto-dominating points and non-dominating points. Moreover, we aim to perform this classification with tunable certainty, i.e., we want to be sure that the points we discard are with high certainty worse than those we classify as Pareto optimal (or those that are still unclassified).

In contrast, popular optimization techniques such as Bayesian optimization and efficient global optimization (EGO) aim to find numerical solutions to a set of objective functions or a single, overall objective function obtained through scalarization. An illustration of a typical problem under Bayesian optimization with expected improvement is shown in Figure 87. In this example, sampling the point with the highest expected improvement is likely not the optimal decision in the long run. However, this point is considered the most (locally) optimal due to the overly greedy (i.e., exploitative) nature of EI. For this reason, other metrics such as “lookahead EI” have been suggested.⁵⁰⁸

It is important to realize that in the limit of many samples, all optimization and active learning techniques will allow us to construct a good Pareto front; however, different techniques will take different paths toward reaching this goal. Active learning techniques are more concerned with the overall information gain needed to improve the classification of optimal vs. non-optimal points, whereas the aforementioned optimization approaches try to balance information gain with exploitation via an acquisition function. In the case of ϵ -Pareto active learning (PAL) the “exploitation” occurs implicitly in the “discarding” classification step.

C.2 DESIGN SPACE

For our design space, we considered 4 monomer types and chain lengths between 16 and 48. Furthermore, we must consider that the reverse sequence equals the forward sequence.

The total number of polymers in our design space is then given by

$$n = \frac{1}{2} \sum_{\substack{i=16 \\ i+=1}}^{48} i^4 = 5.361\,189\,0 \times 10^7. \quad (6)$$

This results in more than 53 million possible sequences.

Enumeration is impossible for so many polymers. For example, assuming an average memory requirement of 62 kB per monomer sequence, the memory footprint would correspond to 3.3 TB. This huge number of polymers also justifies our decision to use DoE as an initial sampling scheme of the design space. Here, we considered increments of two, which results in a smaller design space of

$$n = \frac{1}{2} \sum_{\substack{i=16 \\ i+=2}}^{48} i^4 = 1.406\,675\,2 \times 10^7. \quad (7)$$

C.3 INFLUENCE OF COMPOSITION AND SEQUENCE

Figure 89 shows histograms of the standard deviation of each of the polymer descriptors at a fixed polymer composition and varying sequence. From the distributions, we see that the sequence is secondary compared to the composition of the polymer, as the majority of points fall within 1 standard deviation of the mean.

C.4 COARSE GRAINED MODEL

In DPD for polymers, the force is usually computed as

$$\mathbf{F}_{ij} = \mathbf{F}_{ij}^C + \mathbf{F}_{ij}^D + \mathbf{F}_{ij}^R + \mathbf{F}_{ij}^{\text{spring}}. \quad (8)$$

With soft repulsive force

$$\mathbf{F}_{ij}^C = \begin{cases} a_{ij} \left(1 - \frac{r_{ij}}{r_c}\right) \hat{\mathbf{r}}_{ij} & r_{ij} < r_c \\ 0 & r_{ij} \geq r_c, \end{cases} \quad (9)$$

with repulsion parameter a_{ij} , cutoff radius r_c and unit vector $\hat{\mathbf{r}}_{ij}$.

The dissipative force is given as

$$\mathbf{F}_{ij}^D = -\gamma \left(1 - \frac{r_{ij}}{r_c}\right)^2 (\hat{\mathbf{r}}_{ij} \cdot \mathbf{v}_{ij}) \hat{\mathbf{r}}_{ij}, \quad (10)$$

with friction coefficient γ , the velocity difference between the particles, \mathbf{v}_{ij} , and θ being a random number between 0 and 1.

The random force is computed as

$$\mathbf{F}_{ij}^R = \frac{\sigma \theta_{ij}}{\sqrt{\delta t}} \left(1 - \frac{r_{ij}}{r_c}\right) \hat{\mathbf{r}}_{ij}, \quad (11)$$

with noise parameter σ .

The Frenkel spring force term is

$$\mathbf{F}_{ij}^{\text{spring}} = -K_s (r_{ij} - R_0) \hat{\mathbf{r}}_{ij}. \quad (12)$$

Following Smit and co-workers⁵⁰⁹ we chose the spring constant $K_s = 100k_B T$ and equilibrium distance $R_0 = 0.80$ DPD units, which was found in previous studies to be the first maximum of the pair correlation function of a pure monomer system.⁵¹⁰ All simulations were run using a number density set to 3 and an integration timestep of

0.025.

Español and Warren⁵¹¹ have shown that DPD will sample the canonical ensemble if

$$\gamma = \frac{\sigma^2}{2k_B T}. \quad (13)$$

Here we chose $\gamma = 4.5$.

Table 19: DPD cross interaction parameters. “S” denotes the solvent beads, “S2” denotes the beads belonging to the attractive layer of the surface, “S1” denotes the beads belonging to the repulsive core, and “R”, “Ta”, “Tr”, “W” represent the monomer beads.

bead i	bead j	a_{ij}
S	R	30
S	Ta	27.25
S	Tr	27.25
S	W	20
S	S1	25
S	S2	25
R	Ta	25
Ta	Tr	25
Tr	W	25
R	W	25
R	Tr	25
Ta	W	25
S2	R	15
S2	Ta	15
S2	Tr	20
S2	W	20
S1	R	75
S1	Ta	75
S1	Tr	75
S1	W	75
S2	S1	25

c.4.1 Surface model

The adsorption surface is modeled as face-centered cubic (fcc) lattice structure with an equilibrium bond length of 0.707 and a lattice cell length of $\sqrt[3]{4/3} \sim 1.1$ DPD length units. The surface model consists of an inner and outer layer region: The outer layers are one DPD length unit thick and constitute the attractive surface (S2), whereas the inner-layer (8 DPD length units) represents the repulsive core (S1). Note that the simulations are set up symmetrically and thus contain two outer layers for both sides of the surface. All fcc surfaces consisted of 10,240 DPD beads, corresponding to a total thickness of ~ 10 DPD length units. The bond constant of the lattice was set to $100 k_B T$.

C.5 MOLECULAR SIMULATIONS

c.5.1 Free energy of adsorption

Adsorption free energy simulations were initially prepared by solvating the fcc lattice surface model with 25,000 solvent beads (S) and the single-chain polymer. This yielded a box dimension of approximately $17.6 \times 17.6 \times 37.8$ cubic DPD length units

for each polymer system (3125 systems in total). Simulations were set up using Enhanced Monte Carlo (EMC) and run using the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) code. Standard molecular dynamics (MD) simulations were performed for 1 M timesteps for all systems, prior to running steered MD and umbrella sampling simulations.

Both steered MD and subsequent umbrella sampling simulations were performed using the “ParticleSeparation” collective variable (CV) defined in the Software Suite for Advanced General Ensemble Simulations (SSAGES) code. The CV distance was defined as the distance between the fcc lattice center-of-mass and polymer center-of-mass along the z (normal) dimension. CV distances were set 1 DPD length units apart with minimum and maximum CV values of 1 to 12, respectively. This corresponded to 12 simulations being performed for each polymer system ($12 \cdot 3125$ polymers = 37,500 simulations in total). Steered MD simulations were first performed for 200,000 timesteps using a biasing spring constant of $15 k_B T$ to steer the polymer into the target CV windows. Umbrella sampling was then performed for 1 M timesteps using a spring constant of $15 k_B T$ for each umbrella. The weighted histogram analysis method (WHAM) was used to obtain the final potential of mean force (PMF) as a function of surface-polymer z center-of-mass separation distances. We found that setting the spring constant to $15 k_B T$ with CV separation distances of 1 DPD length unit provided good overlap between CV distance histograms used in WHAM. The adsorption free energy of adsorption (ΔG_{ads}) was taken as the difference in free energy between free energy minimum along the z -dimension and the free energy of the polymer in the bulk phase, i.e., maximum z separation distance of 12. Note that we assume the Helmholtz free energy and Gibbs free energy as approximately equal (i.e., pV contributions are assumed negligible).

c.5.2 Dimer repulsion energy

Dimer repulsion energy simulations were prepared by solvating two identical polymers with 100,000 solvent beads, which corresponded to a box dimension of $32.2 \times 32.2 \times 32.2$ cubic DPD length units. Steered MD and umbrella sampling simulations were performed using the “ParticleSeparation” CV in the x , y , and z dimensions, i.e., radial dimension. CV distances were set 1 DPD length units apart with minimum and maximum CV values of 0 to 11, respectively. This again corresponded to 12 simulations being performed for each polymer system ($12 \cdot 3125$ polymers = 37,500 simulations in total). Similar to the adsorption free energy calculations, steered MD simulations and subsequent umbrella sampling simulations were performed for 200,000 and 1.5 M timesteps, respectively, with biasing spring constants of $15 k_B T$. WHAM was used to obtain the final PMF curve as a function of polymer-polymer radial center-of-mass separation distance. Entropy corrections of $2 \log(r)$ were also applied to the PMF curve. The dimer repulsion free energy repulsive free energy of polymer dimer (ΔG_{rep}) was taken as the difference in free energy between the free energy at CV separation distances of 0 and the free energy at the maximum radial separation distances.

c.5.3 Radius of gyration

Radius of gyration simulations were prepared by solvating a single polymer with 5000 solvent beads corresponding to a box dimension of $11.9 \times 11.9 \times 11.9$ cubic DPD length unit. All simulations were run for 2 M DPD timesteps.

The radius of gyration (R_g) is computed as

$$R_g^2 = \frac{1}{M} \sum_i m_i (r_i - r_{cm})^2, \quad (14)$$

where M is the total mass, r_{cm} the center of mass and the sum is over all beads, using the `gyration` command in LAMMPS. R_g s were output every 1000 steps and averaged over the entire simulation trajectory.

C.6 FEATURIZATION

c.6.1 Polymer representation

Our text-based polymer notation uses the DPD bead types in place of atoms used in standard “SMILES”. Parentheses can be used to reflect branching, however, we only consider linear polymers for the scope of this work.

c.6.2 Features considered in this work

For the linear polymers, we computed the following features from the monomer sequence. The feature names follow the ones we use in the dataset:

- `length`: degree of polymerization (number of beads)
- `head_tail_{bead}`: 1 if bead at head or tail of polymer, 2 if at head and tail, 0 otherwise
- `rel_shannon`: Shannon entropy of the polymer chain relative to the maximum possible entropy for a chain of the same length
- cluster statistics, describing clusters in which the same bead type is repeated:
 - `num_{bead}`: number of clusters for the bead type, relative total the total number of clusters
 - `total_clusters`: total number of clusters in a polymer chain
 - `max_{bead}`: maximum cluster size for the bead type
 - `min_{bead}`: minimum cluster size for the bead type
 - `mean_{bead}`: mean cluster size for the bead type
- `{bead}`: the frequency of a bead type in the polymer chain
- Statistics of the DPD interaction parameters:
 - `total_solvent`: the sum of the DPD interaction parameters of the polymer chain with the solvent
 - `total_surface`: the sum of the DPD interaction parameters of the polymer chain with the surface
 - `std_solvent`: the standard deviation of the DPD cross-interaction parameters of the polymer chain with the solvent
 - `std_surface`: the standard deviation of the DPD cross-interaction parameters of the polymer chain with the surface

c.6.3 Feature selection for the GPR surrogate models

For the final models we used the following features: `num_[W]`, `max_[W]`, `num_[Tr]`, `max_[Tr]`, `num_[Ta]`, `max_[Ta]`, `num_[R]`, `max_[R]`, `[W]`, `[Tr]`, `[Ta]`, `[R]`, `rel_shannon`, `length`. We used the same features for every surrogate model.

C.7 GPR SURROGATE MODEL

We used the GPy²⁸⁰ package to build and train the GPRs and used the limited memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm for hyperparameter optimization.

For optimization of the hyperparameters we performed 20 random restarts with different initializations and used hyperparameters corresponding to the best maximum likelihood solution.

c.7.1 Utility of coregionalized models

Coregionalized models use multi-output kernels, which have the following form:

$$\mathbf{B} \otimes \mathbf{K} = \begin{pmatrix} B_{1,1} \times \mathbf{K}(\mathbf{X}_1, \mathbf{X}_1) & \cdots & B_{1,D} \times \mathbf{K}(\mathbf{X}_1, \mathbf{X}_D) \\ \vdots & \ddots & \vdots \\ B_{D,1} \times \mathbf{K}(\mathbf{X}_D, \mathbf{X}_1) & \cdots & B_{D,D} \times \mathbf{K}(\mathbf{X}_D, \mathbf{X}_D) \end{pmatrix},$$

where \mathbf{K} is a kernel function, \mathbf{B} is regarded as the coregionalization matrix, and \mathbf{X}_i represents the inputs corresponding to the i -th output. \mathbf{B} allows the model to share information between outputs, which would not be possible for two independent kernels; it is hence especially valuable when one only has training data and sparse response data (i.e., missing data for some of the responses) for multiple objectives (see Appendix C.7.3). In case when all objectives are independent one would find $B_{ij} = 0 \forall i \neq j$. However, in the ϵ -PAL algorithm we are interested in learning any regularities that might exist in our design space as fast as possible.

Therefore we investigated how the coregionalized models perform compared to two separate GPR models given our training data. We chose R_g and ΔG_{ads} as targets. For this analysis, we employed the Matérn-3/2 kernel and ICM for coregionalization. To remove the degeneracy in the variance hyperparameter (one in the ICM and another in the Matérn kernel), we constrained the variance of the Matérn kernel.

Two key parameters for the surrogate models in the ϵ -PAL algorithm are the predicted variance (as this will influence how large our rectangles are) and the accuracy of the prediction (as ϵ -PAL will no utility if the model is not predictive). To investigate if a coregionalized kernel can be of use for our problem, we performed a learning curve analysis, as our overall goal for active learning is to predict well with as little training data as possible. To obtain error estimates, we performed the analysis 10 times with different random seeds. In Figure 94, we find that the coregionalized models outperform the two separate ones—notably for the smaller training set sizes, which we would be suitable for the ϵ -PAL algorithm.

In Figure 95, we show the variance as a function of the training set size.

In Figure 96, we show the predictive performance of the GPR models trained with the ϵ -PAL active learning process.

In Figure 97 and 98, we show the SHAP summary plots for surrogate models trained with $\epsilon = 0.01$ and $\epsilon = 0.1$, respectively,

c.7.2 Other model types

In Figure 99, we compare the predictive performance of a rank 1 ICM model, rank 2 ICM model, independent GPR, the neural tangent kernel (NTK), and the neural network Gaussian process (NNGP). The NTK and NNGP were computed using the neural-tangents library.²⁸⁵ The NTK and NNGP were based on the architecture [8, 8, 8] with error function activation. We did not perform hyperparameter optimization for these NN-based models.

For the GPR models, we used a Matérn-5/2 kernel without ARD. We tested the models for predicting the R_g and ΔG_{ads} .

The learning curves indicate that in our case, there is little difference between rank 1 and rank 2 coregionalized models.

c.7.3 Dealing with missing data

For the case study with missing data, we randomly discarded one-third of the simulation results for the dimer repulsion energy. The PyePAL code can deal with this situation with any kind of model, but coregionalized models are particularly suitable as they can exploit correlations between the objectives and hence help with “filling in” the missing measurement. The progress of an active learning experiment in this setting is illustrated in Figure 100.

C.8 PARETO OPTIMAL STRUCTURES IN FEATURE AND PROPERTY SPACE

c.8.1 Objective (property) space

Out of 3125 polymers in our experimental design, 73 are Pareto optimal according to our brute-force simulation results (Figure 101).

c.8.2 Feature space

To visualize feature space, we project the high-dimensional feature space onto two dimensions using PCA (Figure 102, using `scikit-learn`²⁷⁹) and UMAP (Figure 103, using `umap-learn`⁵¹²). We can observe that the Pareto optimal structures do not cluster in one region of feature space but are spread all over the feature space.

C.9 HYPERPARAMETER TUNING FOR THE PAL ALGORITHM

c.9.1 Influence of the initial training set

One crucial assumption for the theoretical bounds to be valid is that the true error is bounded by the estimate provided by the GPR (this can be problematic if the model is overconfident⁵¹³). For this reason, we found empirically that it is practical to initialize the search with a diverse set of about that is large enough that the model is predictive. The minimum number of samples can be estimated using learning curve analysis. The influence of the number of initial points on the performance on our dataset is shown in Figure 104.

To ensure a good sampling of the initial space, we chose the n samples using a greedy MaxMin sampling, initialized with the point closest to the mean of the dataset.

Figure 105 compares greedy MaxMin sampling with initialization based on k -means clustering. We find that the MaxMin sampling typically leads to faster convergence but that the k -means sampling converges to lower errors.

C.10 ϵ -PAL IMPLEMENTATION

c.10.1 Overview of the algorithm

The input of the ϵ -PAL algorithm is the initial design space E , the priors for the GPR models, as well as the hyperparameters ϵ_i and δ . Additionally, we use a scaling parameter (β_{scale}) of the scaling parameter for the hyperrectangle (β_t) as the theoretical value tends to be too conservative. For most of our ϵ -PAL runs, we set the hyperparameters $\delta=0.05$, ϵ_i , and scaled β_t by $1/20$. In our PyePAL package, we allow the user to choose custom schedules for the optimization of the hyperparameters of the GPR, batch size, and exclude high-variance points from the classification step.

For the subsequent discussion, we need to define the following symbols, which mostly follow the notation from Zuluaga et al.:

- design space (E): finite set of points from which we sample
- (ϵ -accurate) Pareto set (P): the solution we aim to find
- set of unclassified points (U): in the first iteration $U_0 = E$
- set of discarded points (D): points for which we can say with high confidence that they are not ϵ Pareto optimal
- using the standard deviation and mean vectors predicted by the GPRs we use the β_t to compute a conservative uncertainty hyperrectangle of point \mathbf{x} ($Q_{\mu,\sigma,\beta}(\mathbf{x})$)
- iterative intersection of the hyperrectangles gives us uncertainty region of point \mathbf{x} ($R_t(x)$): $R_t(x) = R_{t-1} \cap Q_{\mu,\sigma,\beta}(\mathbf{x})$)

It is also useful to compare the concept of Pareto optimality with the one of ϵ -accurate Pareto dominance, assuming a maximization problem.

- Pareto dominance: We say that y dominates y' iff $y \geq y'$, i.e., y is no worse than y' in all objectives and strictly better in at least one objective
- ϵ -Pareto dominance: We relax the definition to $y + \epsilon \geq y'$, which we also write as $y \succeq_{\epsilon} y'$

Note that in contrast to the original implementation, we do not require knowledge of the value ranges of the objectives to compute the uncertainty hyperrectangles. In general, this is not known *a priori*. That is, instead of computing the tolerance as $\epsilon_j r_j$, where r_j is the range of objective j , we use $\epsilon_j \mu_j$ where μ_j is the prediction for objective j . Additionally, this ensures that the tolerance is proportional to the value of the μ_j (and is not inflated/deflated depending on the range). We compare both behaviors in Figure 108. We find the adaptive tolerances converge to lower errors and to also show lower errors in the initial and intermediate iterations. This is particularly pronounced if we do not set the uncertainties of the sampled points to zero but instead use the uncertainties predicted by the GPR.

For the calculation of the hypervolumes we use code from the `nevergrad` library.²⁸⁶ Note that hypervolumes are not used in the algorithm itself but are logged to monitor our algorithm convergence.

Moreover, we implemented the prediction error $\varepsilon(\hat{P}, P)$ ²⁵⁶

$$\varepsilon(\hat{P}, P) = \frac{1}{\|P\|} \sum_i^{\|P\|} \min_{x' \in \hat{P}} \max_{1 \leq j \leq p} \frac{(f_j(x) - f_j(x')) \cdot 100}{r_j}, \quad (15)$$

where \hat{P} is the predicted Pareto front, p the number of objectives, r_j is the range of the values for objective j , and $\|P\|$ the number of points in the set of Pareto optimal points, P .

For more detail on the ε -PAL algorithm, we refer the reader to the original implementation.^{256,257}

INITIALIZATION To initialize the GPRs a few samples from the design space need to be first evaluated. In practice, this can be done by selecting k samples closest to the centroids of a k -means cluster or by using a greedy MaxMin sampling approach (which we initialize with the median or mean).

MODELLING As discussed in section C.7, we use Gaussian processes as surrogate models and the mean and variance of the posterior function to construct hyperrectangles. We decided to add the frequency of hyperparameter optimization of the GPR models as a hyperparameter for the ε -PAL algorithm.

DISCARDING Any point is removed from the unclassified set if its optimistic outcome is ε dominated by the pessimistic outcome of another point. More specifically, we first consider the Pareto pessimistic set $p_{\text{pess}}(P)$ and then the Pareto pessimistic set $p_{\text{pess}}(P \cup U)$, where a Pareto pessimistic set is defined as the set of points \mathbf{x} for which there is no other point \mathbf{x}' such that $\min(R_t(x)) \leq \min(R_t(\mathbf{x}'))$. For this reason, we are guaranteed that the discarding step is safe since there always will be a point that ε dominates the discarded points. This is a key feature that is of particular importance for materials design and discovery.

IDENTIFICATION OF ε PARETO OPTIMAL POINTS A point \mathbf{x} belongs with high probability to the output set of ε -accurate Pareto points if there is no other point $\mathbf{x}' \in P \cup U$ such that $\max(R_t(\mathbf{x}')) \leq_{\varepsilon} \min(R_t(x))$.

SAMPLING In the sampling stage, the next sample is one of the Pareto optimal or unclassified points with the highest uncertainty w_t :

$$w_t(\mathbf{x}) = \max_{y, y' \in R_t(\mathbf{x})} \left\| \frac{y - y'}{\hat{y}} \right\|_2. \quad (16)$$

Note that the algorithm does not sample from the discarded points. To ensure scale invariance, we rescale the uncertainty in each direction by the mean prediction, i.e., we use the coefficient of variation for sampling. Beyond de-biasing the search, this change can have an impact on the performance of the algorithm as shown in Figure 110.

As the choice of the aggregation function (with which the different objectives are combined into one scalar for the sample step) is not unique, we compared the performance of the Frobenius norm (default in the PyePAL package, used for this work) with the mean and median (Figure 111).

In our implementation, we, by default, will use the measured mean and standard deviations instead of the predictions of the surrogate model. From Figure 112, we

see that replacing the GPR uncertainty for the sampled points with zero greatly expedites the convergence.

We chose to not implement sampling methods that require retraining of the models for all potential candidates (e.g., expected error reduction^{514,515}) as those techniques would extremely increase the computational cost of the algorithm (retraining and evaluating the model(s) for every possible new sample, averaged over all possible labels), even though those techniques might mitigate the tendency of uncertainty sampling⁵¹⁶ to sample outliers.

STOPPING The algorithm stops when all points are either discarded or classified as Pareto optimal, i.e., if $U = \emptyset$.

c.10.2 Batch sampling

Batch sampling can be beneficial if simulations or experiments can be parallelized and when a sequential scheme is too time-consuming. The original ϵ -PAL scheme samples only one sample per iteration. In this work, we did not employ batch sampling. However, in our PyePAL package, we allow the user to perform ϵ -PAL in batch mode. We use a greedy approximation and sample the n next best samples according to the selection criterion rather than just sampling one point. Note that the greedy approximation lowers the efficiency of the exploration of the space.

c.10.3 Multiple ϵ

For some applications, it is often preferred to have the different tolerances for ϵ on the Pareto front for each objective. To account for such flexibility, we use one ϵ_i per dimension i .

c.10.4 Theoretical guarantees

Zuluaga et al.²⁵⁶ showed that the ϵ -PAL algorithm comes with some guarantees for the quality of the solution. For that reason, one assumes that the functions which are modeled with the GPRs are arbitrary functions from the reproducing kernel Hilbert space (RKHS) with some associated kernel k . Additionally, the noise of the samples is assumed to have zero mean conditioned on the history and to be bounded by σ . For an appropriate choice of hyperparameter β_t , Zuluaga et al. proved that an ϵ -accurate Pareto front can be found in a bounded number of iterations with probability $1 - \delta$, where δ is also a hyperparameter that can be specified by the user.

c.10.5 Limitations

It is well known that kernel methods tend to need stringent feature selection.²³⁹ In case one only has access to a high-dimensional feature space with noisy data, the GPR models might have a too low predictive performance for a decent convergence of ϵ -PAL. Future work could investigate Monte-Carlo dropout-based surrogate models.⁵¹⁷

Furthermore, ϵ -PAL, operates on a finite design space, i.e., it will not find a design that is not in the set of possible designs it is provided with as input. Therefore, any continuous design space needs to be discretized.

Additionally, since we require the returned points to be ϵ -Pareto optimal, this approach can require many iterations until the first point is classified as ϵ -Pareto optimal if the predicted variance stays high even after many iterations.

c.10.6 Sensitivity of the hypervolume error to the choice of the reference point

Since the reference point for the hypervolume calculation is a user-defined parameter, we explored a range of different settings. For the reference points, we considered the minimum of our design space (sampled using the DoE), some intermediate point $(-5, -5, -5)$, and a point orders of magnitude larger than the objectives $(-1000, -1000, -1000)$. In Figure 113, we compared the convergence behavior to the median of 100 random explorations of the design space.

C.11 INVERSE DESIGN

We use the term “inverse design” to refer to finding a valid polymer that maximizes the outputs of our models.⁴²⁶

c.11.1 Surrogate model

To be able to perform practical inverse design, we cannot use all the features used in this work. For example, the Shannon entropy feature, provided a length and set of characters, is partially invertible but would require a complex constraint for our optimization algorithm. This would make the search inefficient.

To avoid this issue, we trained GBDTs models with reduced feature sets to predict the predictions of the GPR models. For those surrogate models, we optimized the hyperparameters with Bayesian optimization. The details can be found under sweep ids f2cteo9b, 704xtppt, 1vwsrp8b on the wandb platform). Note that using surrogate models that are trained on the predictions of other models is a commonly used technique to interpret models.^{334,518}

c.11.2 Genetic algorithm

We also attempted to use particle swarm optimization (PSO) as implemented in the `pyswarm`⁵¹⁹ package but found better results with GAs, for which we used the `geneticalgorithm` Python package.²⁹¹ The main advantage of GA over PSO for our optimization problem is that GA allows for a more natural treatment of mixed datatype (integer and real) optimization.

Since not all possible features correspond to valid polymers and we are mostly interested in novel polymers, we used the following fitness function

$$\mathcal{L}(\mathbf{x}) = -10 \frac{\hat{y}(\mathbf{x})}{\bar{y}_{\text{train}}} + \text{CP}(\mathbf{x}) + \text{NIVP}(\mathbf{x}) + \alpha \text{NP}(\mathbf{x}), \quad (17)$$

where \hat{y} is the prediction of the model. CP and NIVP are penalty terms that penalize structures with unphysical cluster size features and those which are not invertible, respectively. The penalties have the following form

$$\text{CP}(\mathbf{x}) = \begin{cases} \max_{\text{bead_type}} > \text{length} \cdot \text{bead} & +30 \\ \text{else} & 0 \end{cases} \quad \forall \text{bead}, \quad (18)$$

where we used the feature notation from section C.6.2, and

$$\text{NIVP}(\mathbf{x}) = \begin{cases} \text{at least one valid monomer seq. generated} & 0 \\ \text{else} & 50, \end{cases} \quad (19)$$

where we use the algorithm described in section C.11.3 to iteratively attempt the mapping from features to a bead sequence.

The last term in eq. 17 is a penalty term that increases the loss for structures that are similar to the ones from our database to encourage the GA to explore new areas of chemical space:

$$\text{NP}(\mathbf{x}) = \max\left(\bar{y}_{\text{train}} \left(\overline{\min \|\mathbf{x}_{\text{train},i} - \mathbf{x}_{\text{train}}\|} - \min \|\mathbf{x} - \mathbf{x}_{\text{train}}\|\right), -\bar{y}_{\text{train}}\right). \quad (20)$$

To consider a wide balance between exploration and exploitation, we swept through $\alpha = \{0, 0.1, 0.5, 1, 2, 10, 20, 50, 100\}$ with multiple random restarts for each α .

We considered the feature bounds listed in Table 20.

Table 20: Feature bounds for the GA.

feature	lower bound	upper bound	type
length	16	48	int
max_{bead}	0	36	int
{bead}	0	1	real

For the elitist GA we used the hyperparameters listed in Table 21.

Table 21: Parameters used for the GA.

parameter	value
restarts	3
maximum number of iterations	300
elite ratio	varied {0, 0.01, 0.05, 0.07}
population size	300
mutation probability	0.1
crossover probability	0.8
parents portion	0.1
crossover type	uniform
early stopping after	500 iterations without improvement

c.11.3 Mapping back to valid monomer sequences

For our promising solutions, we enumerated possible monomer sequences.

To map back to physically valid bead sequences, we round the bead numbers and degree of polymerization found in GA to integers and calculate the maximum size of the clusters to capture the topological features of the polymer (pool of beads and clusters).

To efficiently evaluate if any permutation of the given characters can build a valid monomer sequence, we use a backtracking algorithm.⁵²⁰ This algorithm sequentially evaluates if adding a character from the pool can still provide a valid monomer sequence, based on the constraints on the number of beads and the maximum size of the clusters. If the addition is successful, we pop this character from the pool and continue recursively calling the function until the pool of candidate beads is empty.

Since none of the predictions we made for the polymers we found in the “inverse design” step dominated a point from the Pareto front we found in the subspace sampled with DoE, we did not perform any additional simulations.

In Table 22, we compare hypervolumes found with the GA, and ϵ -PAL on the subspace sampled with DoE.

Table 22: Hypervolume of design space sampled with DoE and Pareto front found using the GA.

reference point	hypervolume original	hypervolume GA	PyePAL ($\epsilon=0.01$)
minimum (1.46, 3.04, 1.61×10^{-3})	268	166	257
(-5,-5,-5)	1536	1283	1515
(-1000, -1000, -1000)	1 026 829 209	1 025 214 020	1 026 813 775

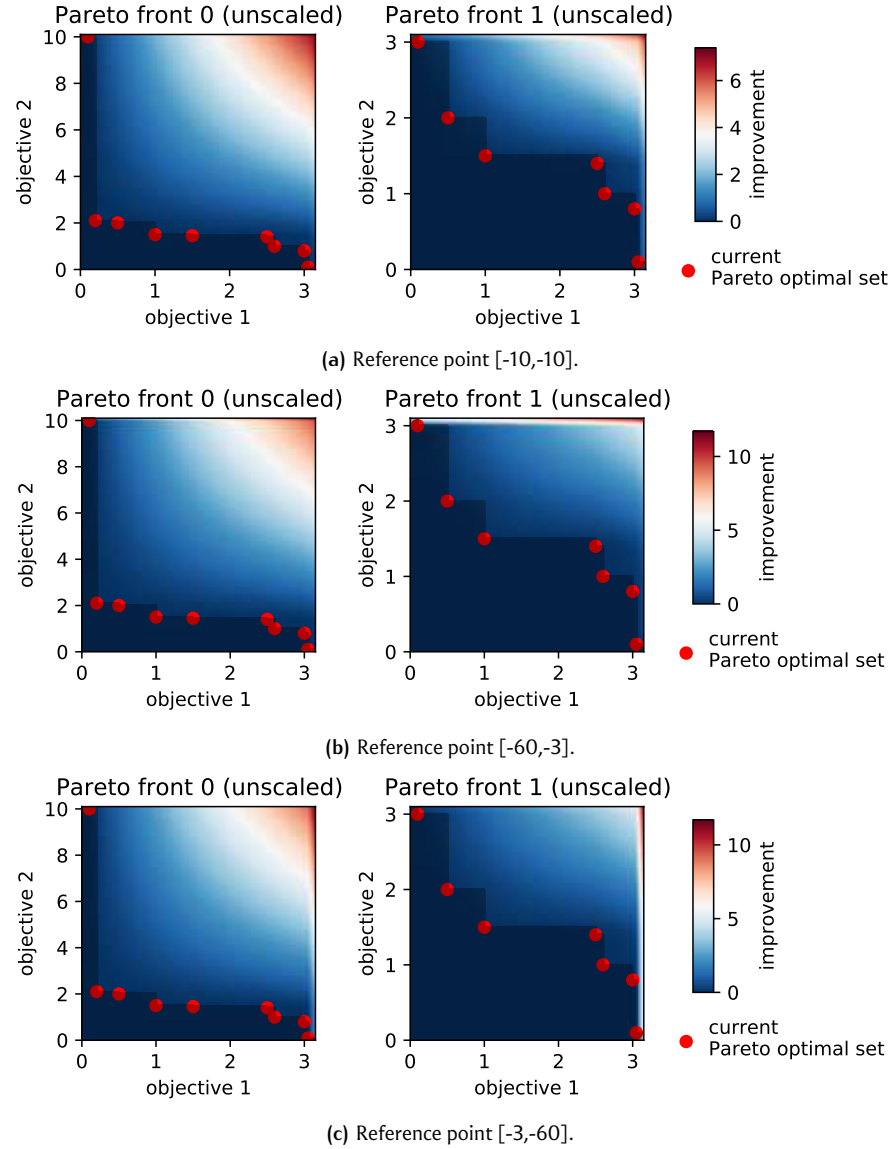


Figure 86: Improvement function landscape based on the improvement in hypervolume, computed with different reference points for the integration of the area below the Pareto frontier. The color coding indicates the value of the improvement function in each grid point. In a Bayesian optimization scheme, we would choose the sample with the highest value of the improvement function, assuming that the variance is equal for all points. In a materials discovery setting, however, there will not be a material on each grid point. A material with a certain combination of objectives might not exist, wherefore there would be blank spots in the figure.

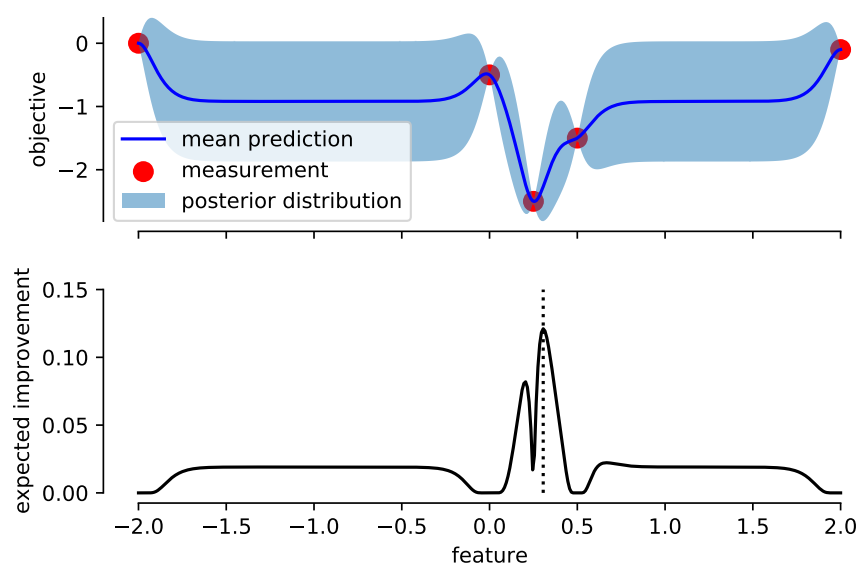


Figure 87: Expected improvement and predictive (posterior) distribution for a GPR model (Matérn-5/2 with automatic relevance determination (ARD)) on noisy samples from the Branin function.⁵⁰⁴ Dotted line indicates the feature value we would sample next based on the maximum value EI. In this particular case, it seems more intuitive, though, to take a sample in the undersampled regions with high uncertainty. Figure based on illustrations in Wu and Frazier.⁵⁰⁸

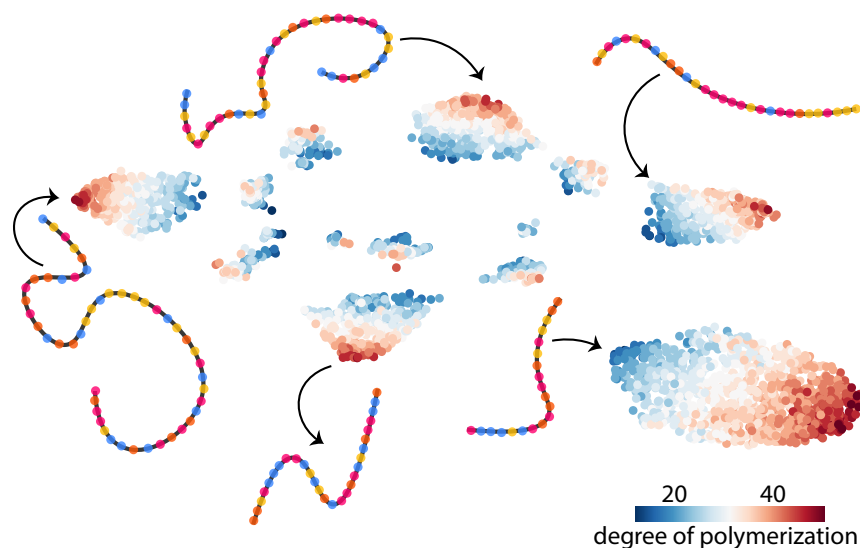


Figure 88: Two-dimensional projection of our polymer design space. We used the UMAP technique to project our design space, which we sampled using DoE, onto two dimensions. Points are colored according to the degree of polymerization. Cartoons illustrate the composition of some copolymers.

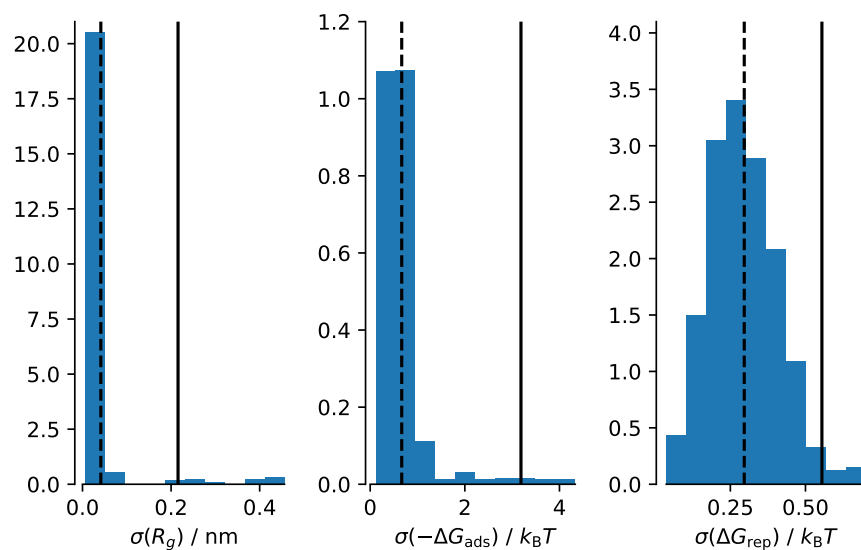


Figure 8g: Histograms show the standard deviation of the descriptors for fixed polymer composition and varying sequence. The black solid line shows the standard deviation between the means of the descriptors for fixed composition. The dashed vertical line gives the mean standard deviation for fixed composition (means of the histograms).

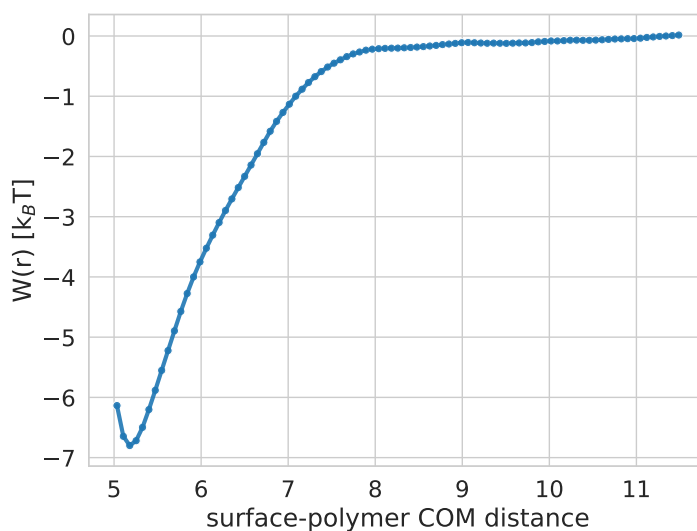


Figure 9g: Potential of mean force (W) as a function of surface and polymer center-of-mass separation distance along the Z normal direction. ΔG_{ads} is taken as the difference in free energy between the bulk and minimum free energy.

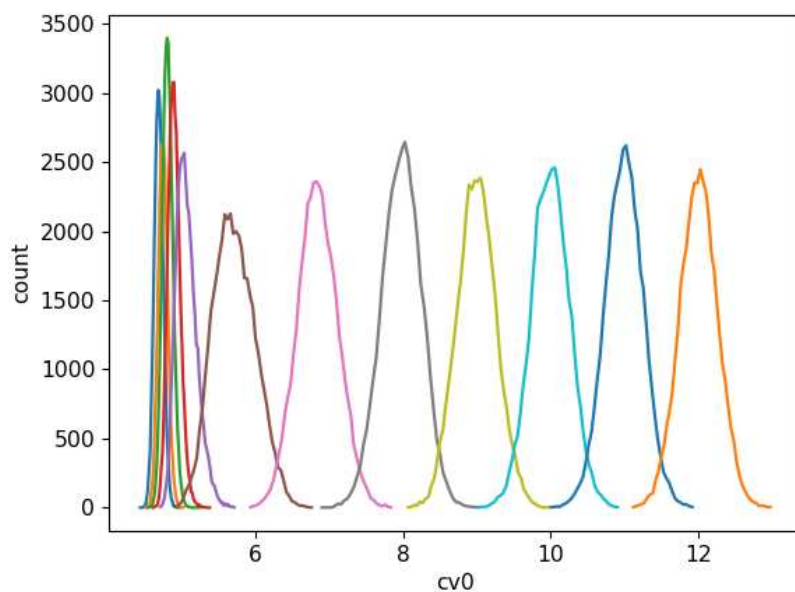


Figure 91: Umbrella sampling histograms obtained as a function of surface-polymer COM separation distance ($cv0$). Target CV separation distances were set to 1 DPD unit apart.

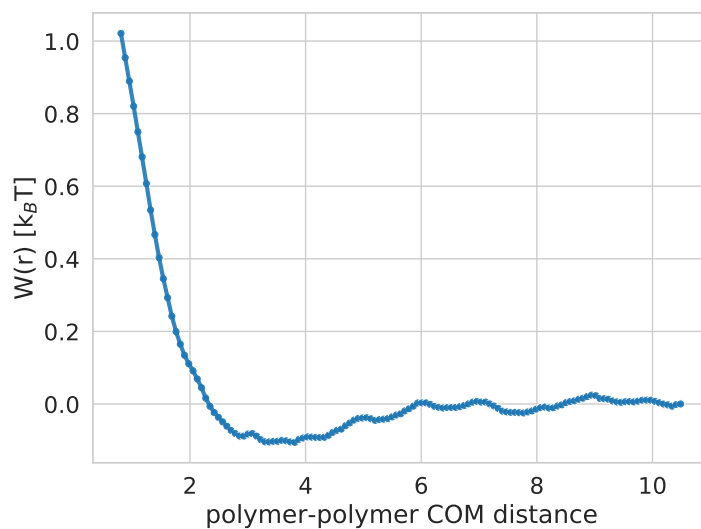


Figure 92: Potential of mean force (W) as a function of polymer and polymer center-of-mass separation distance along the radial (r) direction. ΔG_{rep} is taken as the difference in free energy between those at the maximum and minimum separation distances.

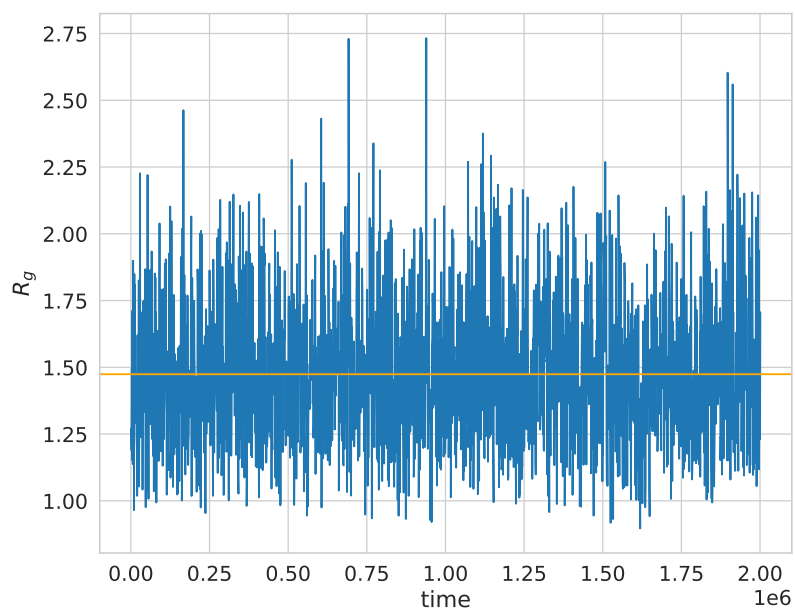


Figure 93: Radius of gyration R_g as a function of simulation timestep. $\langle R_g \rangle$ is taken as the average R_g over the entire simulation trajectory as indicated by the horizontal orange line.

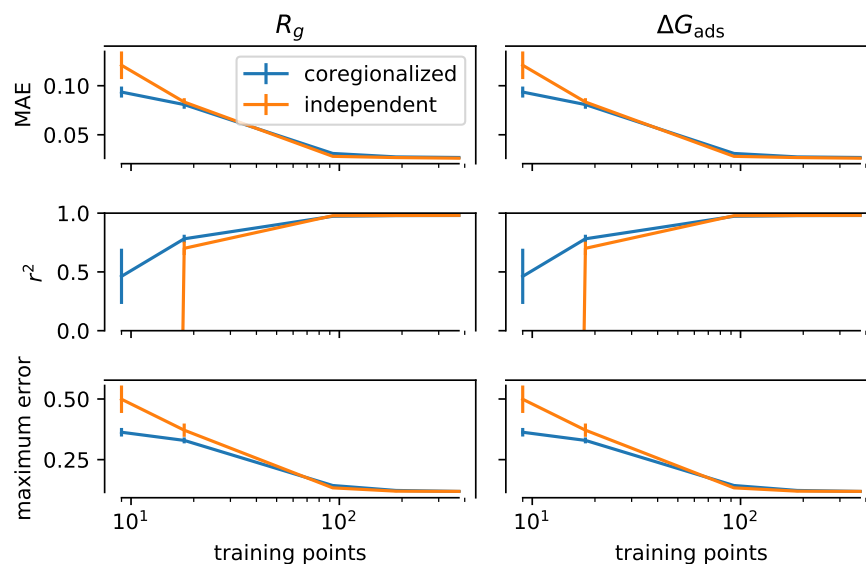


Figure 94: Learning curves for models with coregionalized kernel and independent models.

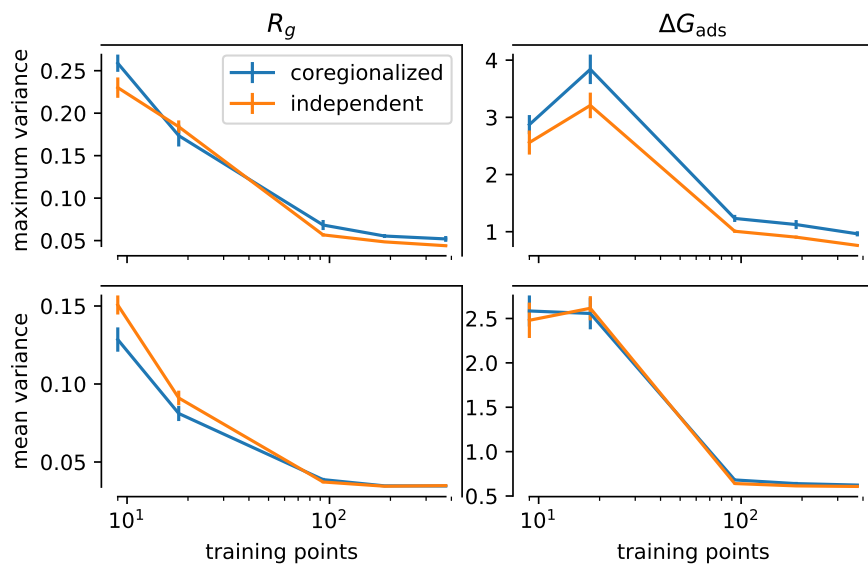


Figure 95: Predicted variance as a function of the training set size for models with coregionalized kernel and independent models.

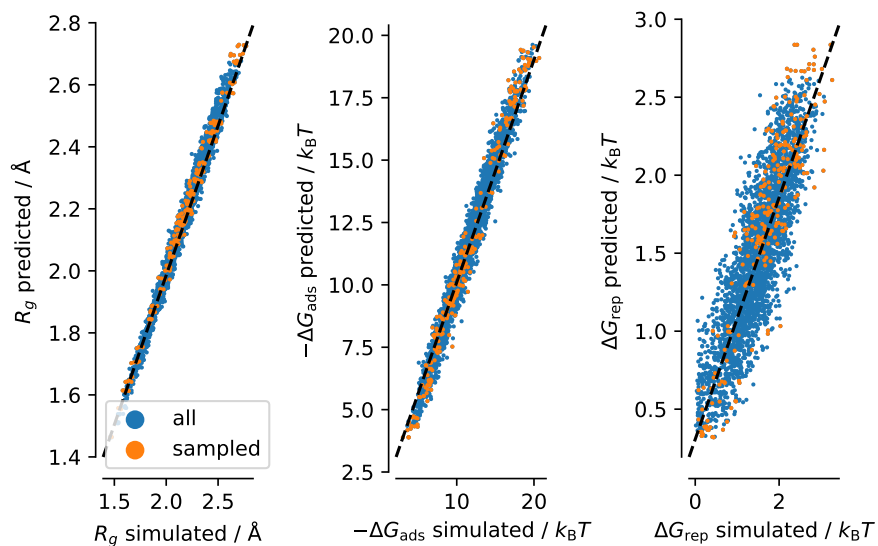


Figure 96: Predictive performance of the models trained with the ϵ -PAL active learning process.

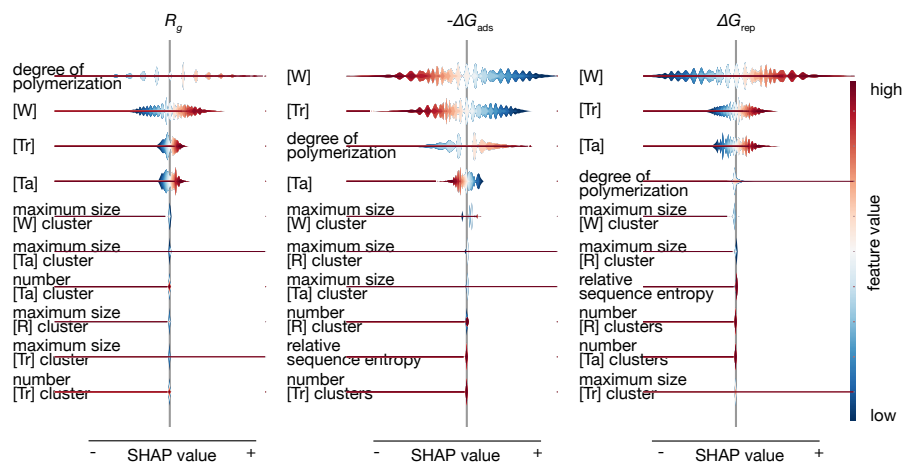


Figure g7: SHAP summary plot for a surrogate model (ICM, Matérn-5/2 kernel) trained over the course of a ϵ -PAL run with $\epsilon = 0.01$, $\delta = 0.05$, $\beta_{\text{scale}} = 0.05$.

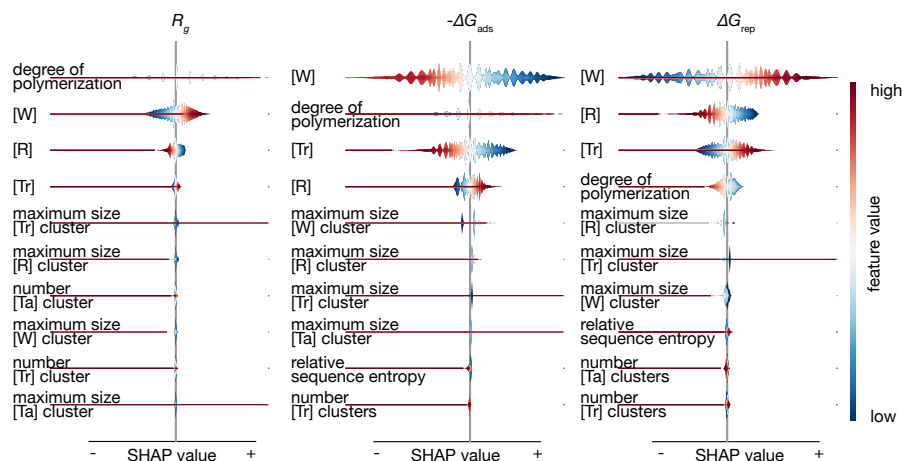


Figure g8: SHAP summary plot for a surrogate model (ICM, Matérn-5/2 kernel) trained over the course of a ϵ -PAL run with $\epsilon = 0.1$, $\delta = 0.05$, $\beta_{\text{scale}} = 0.05$.

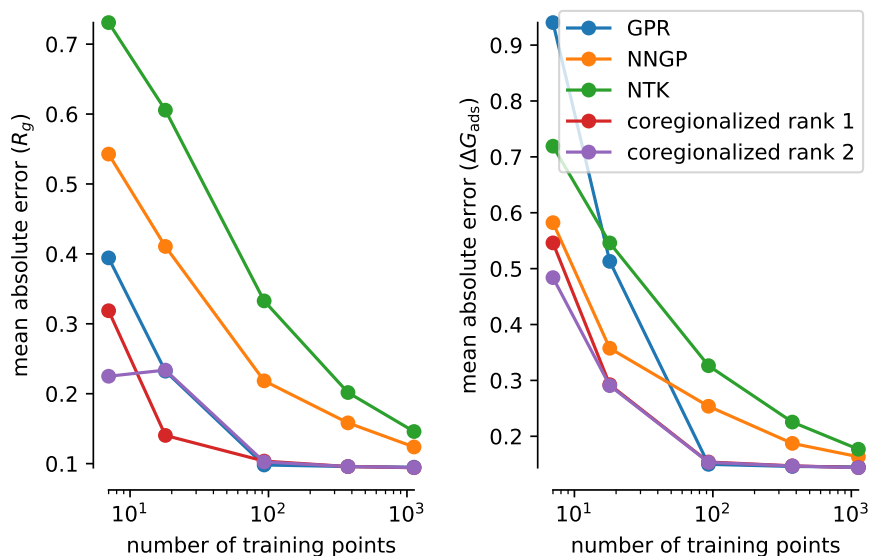


Figure 99: **Learning curves for different model types.** All GPR models were built using Matérn-5/2 kernels without ARD. The NNGP and NTK were built for a small, three-layer NN.

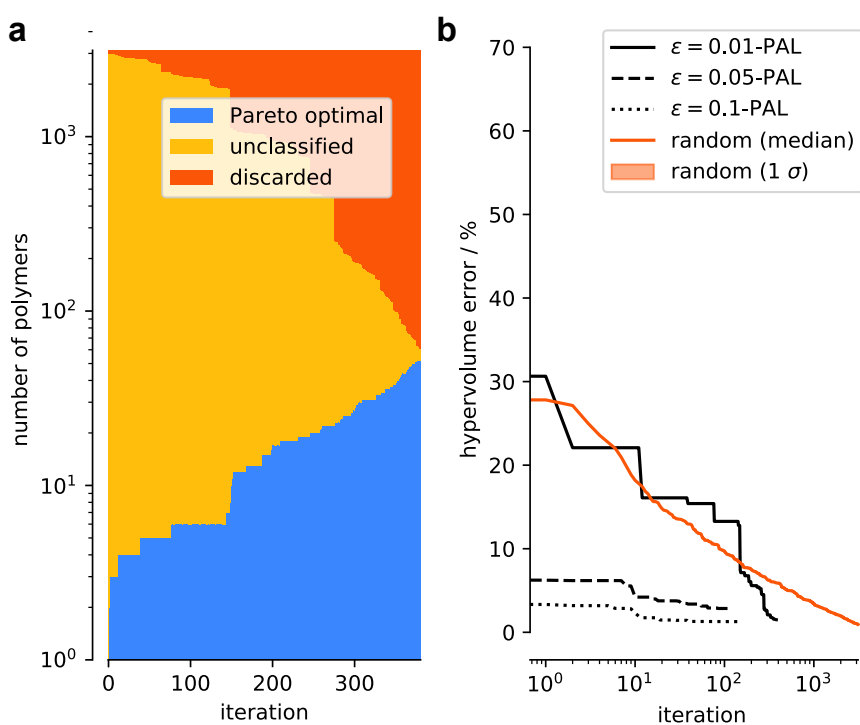


Figure 100: **Classified points and hypervolume error as a function of the number of iterations.** Using ICM with Matérn-5/2 kernel. Hypervolume error for random search (with all data present, i.e., no missing outputs) is shown for comparison. All search procedures were initialized using the same set of initial points but vary substantially after only one iteration step. The hypervolume reference point for this figure is $(-5, -5, -5)$.

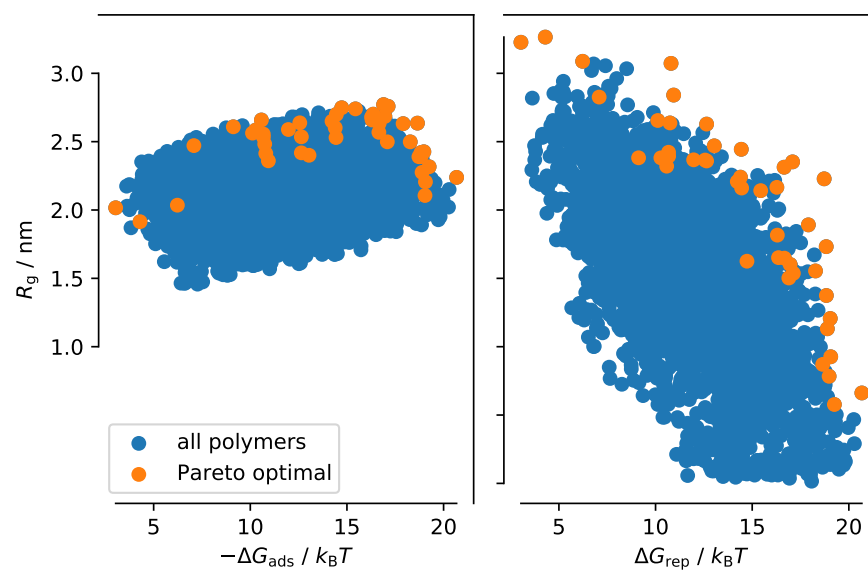


Figure 101: Overview of the design space and the Pareto optimal points in this design space.

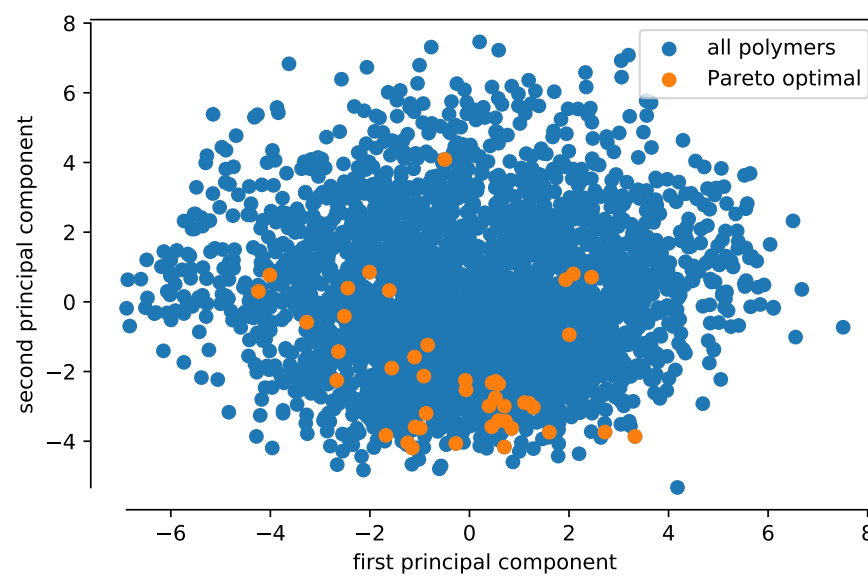


Figure 102: Projection of the feature space onto two dimensions using PCA.

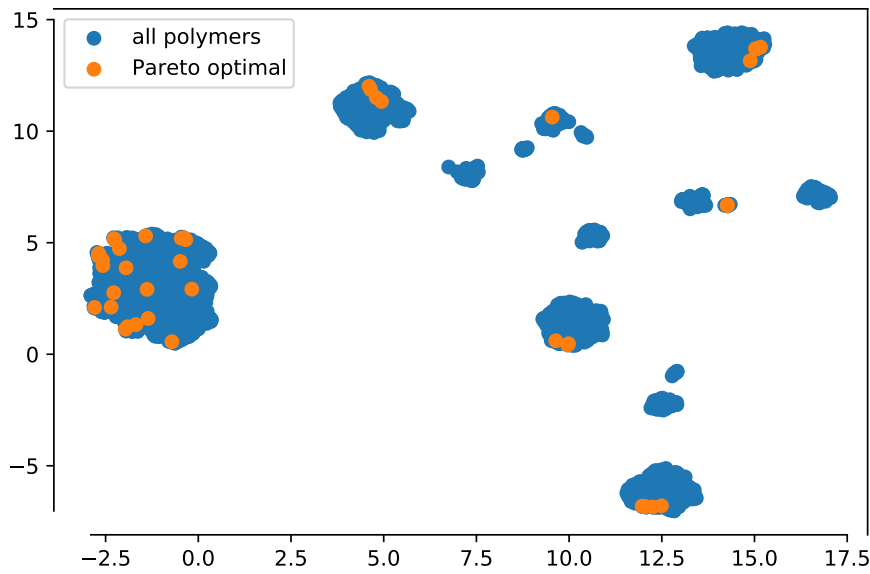


Figure 103: Projection of the feature space onto two dimensions using UMAP.

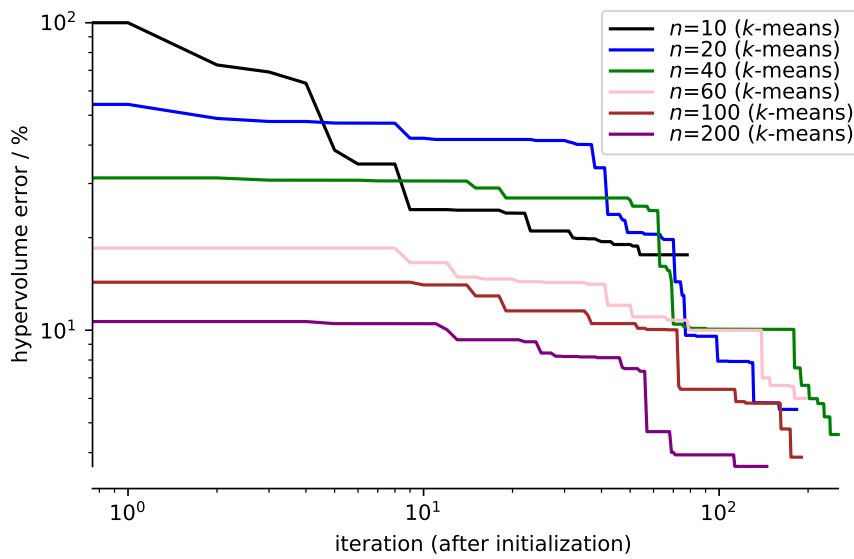


Figure 104: **Influence of the number of initial points.** Hypervolume error as a function of the number of initial points. We left $\epsilon = 0.05$, $\delta = 0.05$, $\beta_{\text{scale}} = 0.05$ fixed and sampled using k-means sampling. Note that a low number of initial samples (e.g., $n = 10$) can lead to unreliable results. This can be the case if the surrogate model is non-predictive and overconfident, causing the misclassification of points early in the search. For this reason, the PyePAL package warns users when the cross-validation error is greater than the variance of the model. Moreover, the learning curves (Figure 99) indicate that the models have a large generalization error for $n \ll 60$. Hypervolumes were calculated using the nadir point as our reference point.

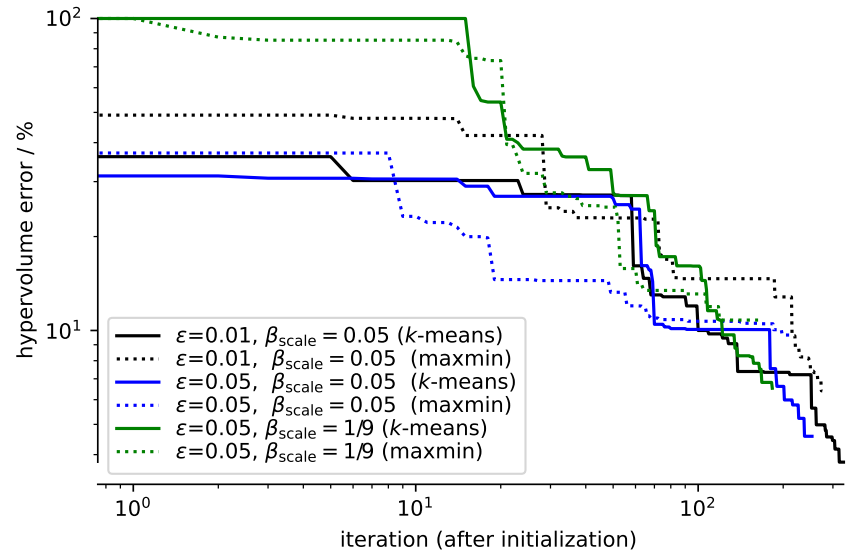


Figure 105: Influence of the sampling method used to create the initial set. Hypervolumes were calculated using the nadir point as our reference point.

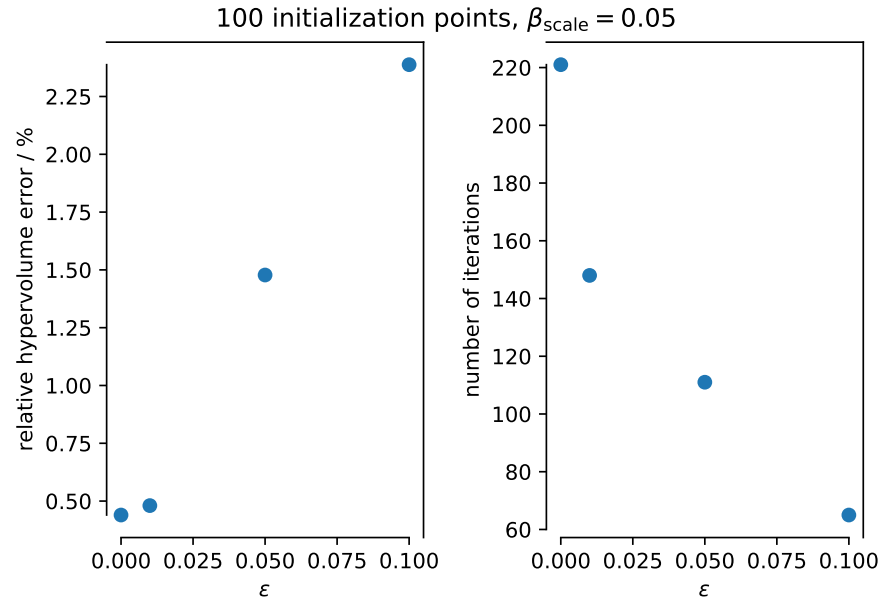


Figure 106: Relative hypervolume errors and total number of iterations of the ϵ -PAL algorithm as a function of ϵ ($\epsilon_i = \epsilon \forall i \in \{0, 1, 2\}$).

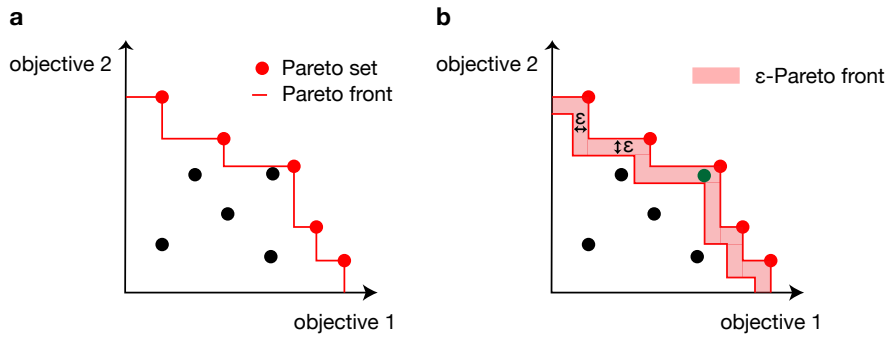


Figure 107: **Pareto vs. ϵ -Pareto front for two objectives.** **a** illustrates the concept of the Pareto set, i.e., the set of maximal points, and the Pareto, whereas **b** shows a ϵ -Pareto front. ϵ Pareto optimality would also be given if the Pareto set includes the green point instead of the neighboring red one.

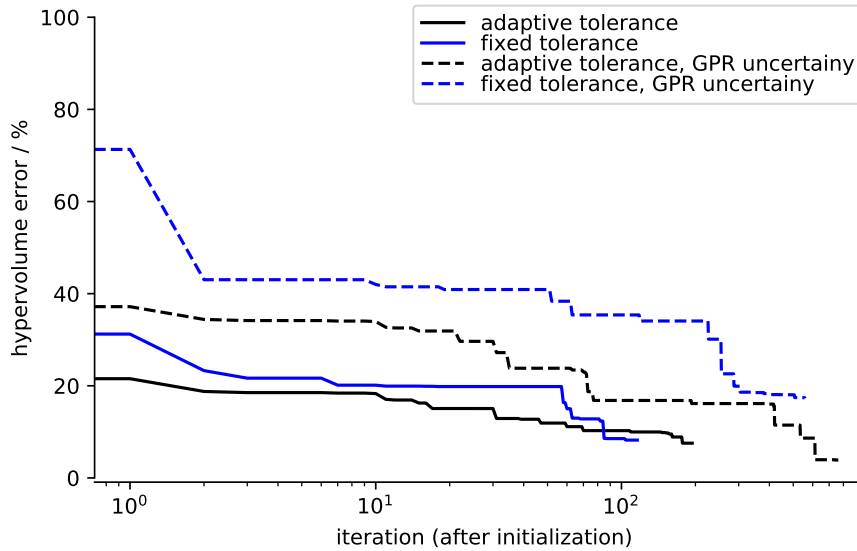


Figure 108: **Fixed tolerances vs. adaptive tolerances.** To expedite this experiment we used a \lg_2 spaced schedule for hyperparameter optimization (in contrast to the linearly space schedule used in the rest of this work). We left $\delta = 0.05$ and $\beta_{\text{scale}} = 0.05$ fixed, used a Matérn-5/2 kernel without ARD, and initialized with 40 points sampled using greedy farthest point sampling. Hypervolumes were calculated using the nadir point as our reference point.

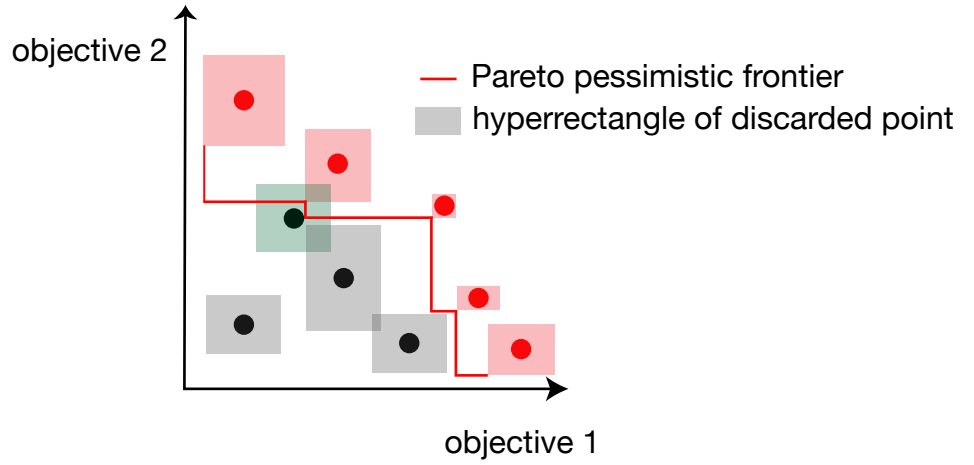


Figure 109: Illustration of the set of Pareto pessimistic points. In the discarding step, we would keep the point with the green uncertainty region $R_i(x)$ as we cannot say with certainty that it is lower than the Pareto pessimistic p_{pess} front but will discard all points with gray hyperrectangles. In the discarding step, we build two different Pareto pessimistic fronts. First, only from the points we already classified as ϵ -Pareto optimal. Then, followed by ones where we consider the union of ϵ -Pareto optimal and unclassified points.

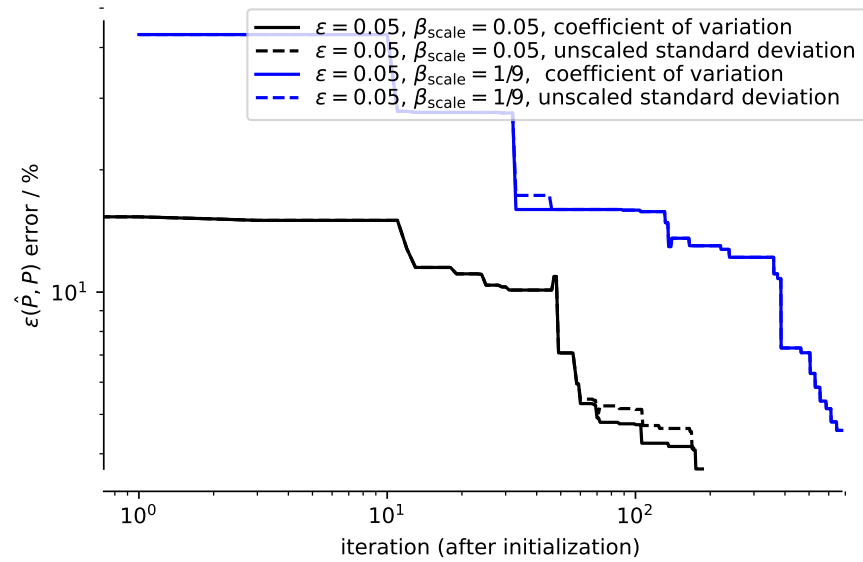


Figure 110: Influence of scaling the variance on the performance of the algorithm. Using the coefficient of variation instead of the unscaled uncertainty marginally improves the performance in our test cases.

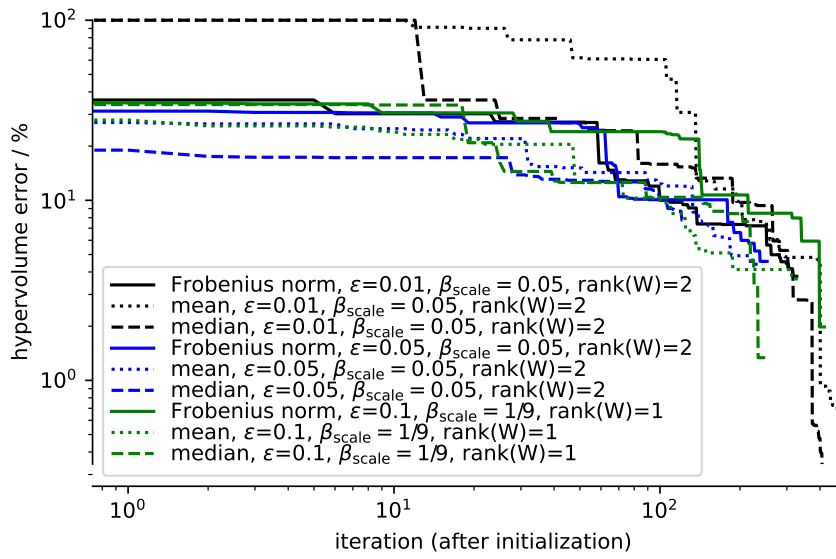


Figure 111: Influence of the aggregation function. We observe that the Frobenius norm leads to faster termination of the search, e.g., compared to median aggregation, which leads to lower final hypervolume errors. Hypervolumes were calculated using the nadir point as our reference point.

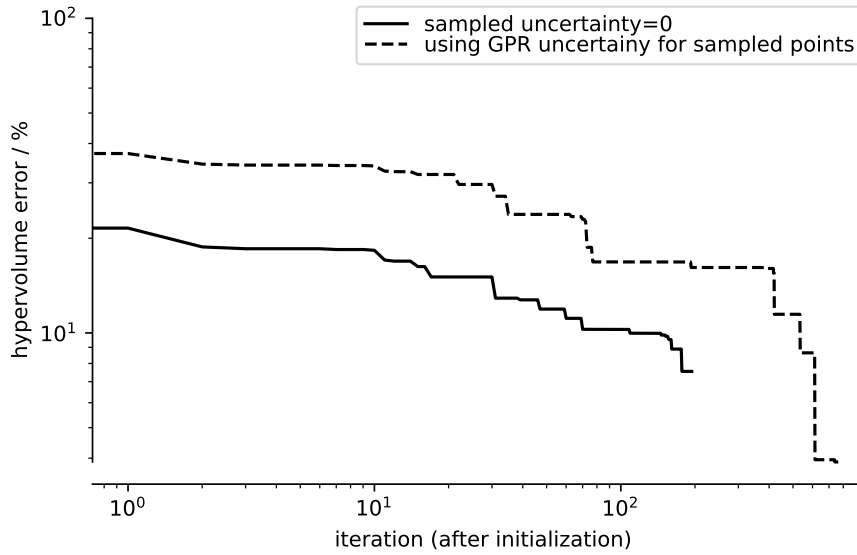


Figure 112: Replacing the uncertainty with the measured uncertainty (here assumed to be zero). To expedite this experiment we used a \lg_2 spaced schedule for hyperparameter optimization (in contrast to the linearly space schedule used in the rest of this work). We left $\delta = 0.05$, $\epsilon = 0.05$, and $\beta_{\text{scale}} = 0.05$ fixed, used a Matérn-5/2 kernel without ARD, and initialized with 40 points sampled using greedy farthest point sampling.

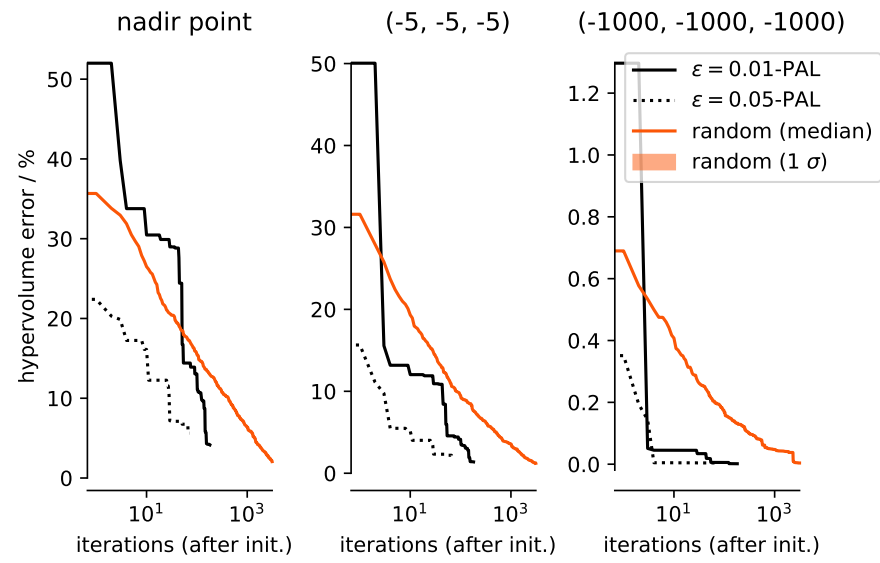


Figure 113: Convergence behavior compared to random search for different hypervolume reference points.

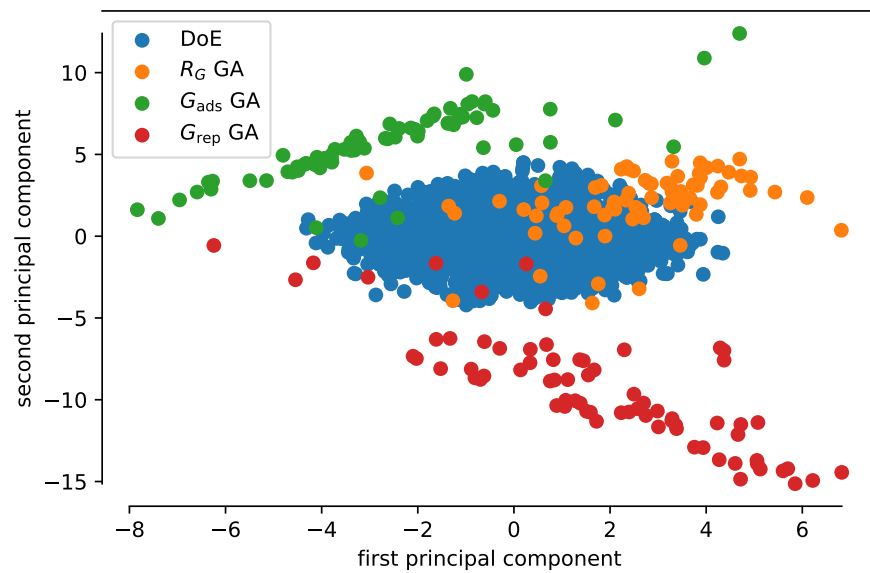


Figure 114: Projection of the generated feature sets on the first two principal components of the database.

D | SUPPORTING INFORMATION FOR "USING COLLECTIVE KNOWLEDGE TO ASSIGN OXIDATION STATES "

D.1 EXPLORATORY DATA ANALYSIS AND TRAINING SET

The data flow used in this work is schematically summarized in Figure 115.

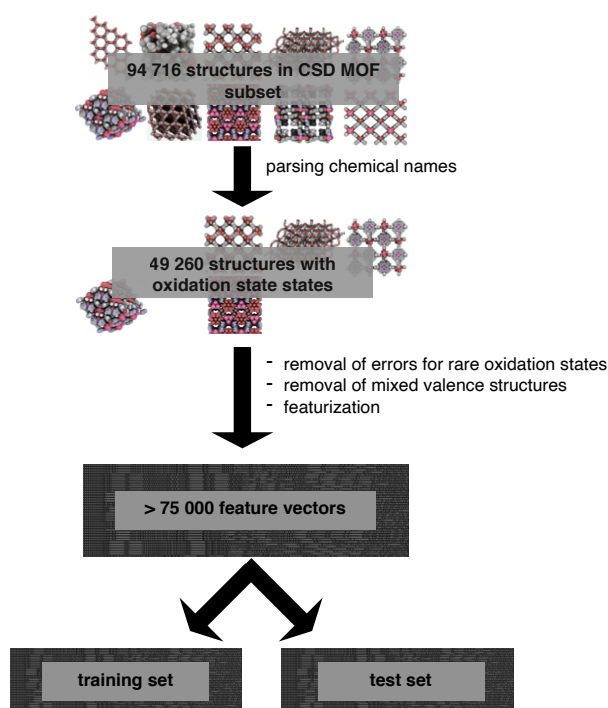


Figure 115: **Dataflow.** High-level overview over the path from structures from the CSD database to the training set.

D.1.1 Construction of the training set

Given that manual annotation of oxidation states of metal centers for a large collection of structures is unfeasible, we decided to parse the available information in the CSD. Currently, there is no searchable field for oxidation states in the CSD, wherefore we used a regular expression to extract the oxidation states from the chemical names (e.g., catena-(tris(μ -4,4',4'',4'''-(porphyrinato-5,10,15,20-tetrayl)tetrabenzoato)-octakis(μ -hydroxo)-octakis(μ -oxo)-dodeca-aqua-hexacarbonyl-dodecahydroxy-tri-cobalt(ii)-dodeca-zirconium(iv)). The Roman literals were then converted to Arabic numerals for the machine learning pipeline. In recent efforts of the CSD, the oxidation states

have been validated.^{305,316}

Private communication by the CSD confirmed that all oxidation states in the CSD have been added by an editor based on the chemistry in the structure and/or with reference to the paper if unclear.

The parsing functionality is implemented in the `oximachine_featurizer` Python package (DOI 10.5281/zenodo.3567274).

D.1.2 Definition of MOF used for this study

In this work, we follow the algorithmically implementable definition of MOFs that was put forward by Moghadam et al.¹¹⁶. This definition is based on seven rules for substructure search that describe the bond between a metal and an organic compound, following the work from Goldsmith et al.⁵²¹. Additionally, Moghadam et al.¹¹⁶ require that the keyword *catena* is in the name, i.e., that the structure is polymeric. Note that these rules do not directly capture the “potential porosity”, which is an element of the IUPAC provisional recommendations⁵²² for the definition of MOFs). Therefore, our training and test set contain structures that chemists might not classify as MOF but rather, e.g., polyoxometallates (see representative structures in Figure 119 and 120, in Appendix D.1.7). Similar observations were recently reported by Chen and Manz⁵²³.

D.1.3 Visualization of the metal distribution

To analyze whether the distribution of elements in the MOF subset of the CSD is different from the distributions of elements in the CSD overall, we created Figure 116. As an additional point of reference, we also plot the distribution of metals in the CoRE-MOF database, which is a subset of the CSD that was selected in 2014 using slightly different criteria. Primarily, it considers in its selection only 3D structures that are porous to hydrogen, i.e., with a pore limiting diameter larger than 2.4 Å. The CoRE-MOF contains ca. 5000 structures, and has been widely used in computational studies.^{134,524} To avoid comparability problems due to varying cell sizes, we raise the count for a particular metal by one if we find it in the structure, independent of how many metal nodes are in that structure.

We observe that the distribution of CSD MOF subset is generally closer to the one of the CSD overall than the distribution of the CoRE-MOF database. Both the CSD MOF subset and the CoRE-MOF database are enriched in Zn, Cu, Cd, Co and deprived in Fe, Ru, Pt, Tl, Au, Rh, Re, Ir, Os. This, on the one hand, shows opportunities for the MOF community, like gray areas in chemical space, and, on the other hand, indicates that some aforementioned metals are cases in which one might expect the model to transfer less well to new chemistry.

D.1.4 Distribution of oxidation states

Before the modeling, we performed exploratory data analysis (EDA) focusing on the frequency of oxidation state annotations in the CSD and the distribution of oxidation states (cf. Table 23).

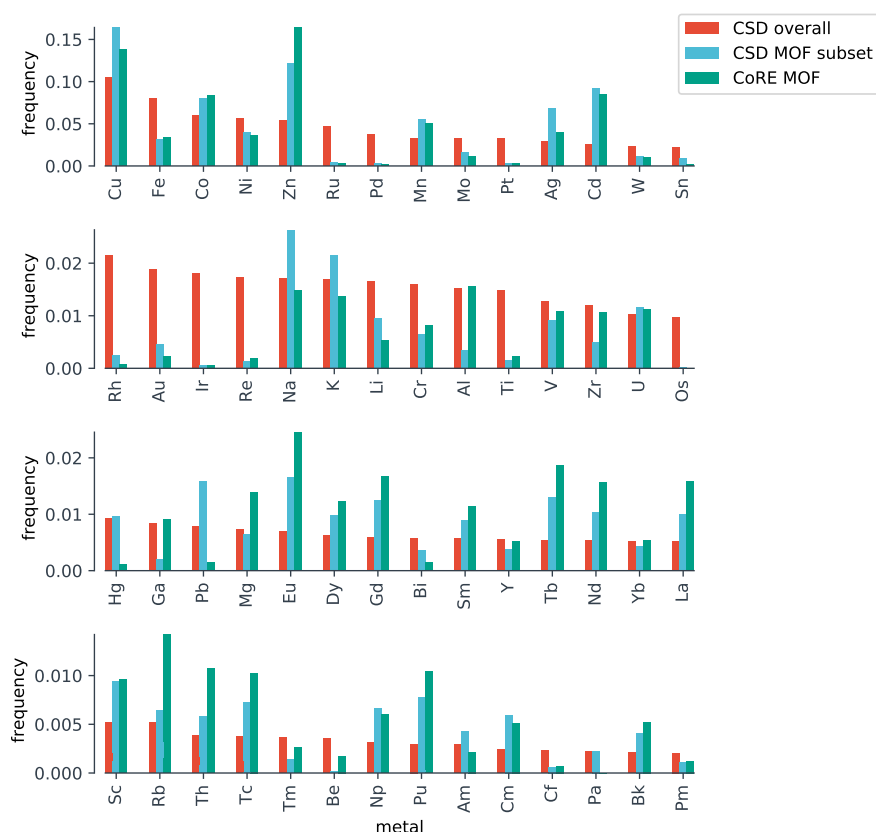


Figure 116: Metal frequencies. Metal distributions in different subsets of the CSD.

Table 23: Distribution of metals and oxidation states in the CSD overall and the MOF subset of the CSD, sorted by count in the full CSD. Frequencies of occurrence are given in percentage.

	CSD overall			MOF subset		
elem.	count	freq.	oxidation states (freq.)	count	freq.	oxidation states (freq.)
Cu	54985	16.9	I (23.2), II (76.4), III (0.4)	11440	20.8	I (24.2), II (75.8), III (0.0)
Ni	26004	8.0	I (1.0), II (95.6), III (3.3), IV (0.2)	2493	4.5	I (0.1), II (99.0), III (0.9)
Co	25042	7.7	I (2.1), II (64.0), III (33.8), IV (0.0), V (0.0)	4816	8.7	I (0.1), II (94.2), III (5.7)
Zn	22672	7.0	I (0.1), II (99.9)	6538	11.9	I (0.0), II (100.0)
Fe	19473	6.0	I (1.1), II (54.6), III (43.7), IV (0.6), V (0.0)	1986	3.6	II (68.4), III (31.5), IV (0.1)
Pd	18539	5.7	I (1.0), II (98.0), III (0.4), IV (0.6), V (0.0)	101	0.2	I (1.0), II (97.0), III (1.0), IV (1.0)
Pt	14695	4.5	I (0.6), II (85.9), III (2.0), IV (11.5), VI (0.0)	180	0.3	II (87.8), III (3.9), IV (8.3)
Mn	14424	4.4	I (4.2), II (64.8), III (26.6), IV (4.0), V (0.4), VI (0.0), VII (0.0)	3578	6.5	I (0.1), II (86.9), III (12.5), IV (0.4), V (0.1)
Ag	11591	3.6	I (98.9), II (0.6), III (0.4)	4621	8.4	I (99.7), II (0.3)
Cd	10814	3.3	I (0.0), II (99.9), III (0.0)	5490	10.0	I (0.0), II (99.9), III (0.1)
Ru	10573	3.2	I (1.3), II (83.1), III (12.2), IV (2.8), V (0.1), VI (0.6)	174	0.3	I (1.1), II (66.7), III (31.6), IV (0.6)
Au	6999	2.1	I (73.7), II (1.8), III (24.5), IV (0.0)	286	0.5	I (90.6), III (9.4)

Continued on next page

Table 23: Distribution of metals and oxidation states in the CSD overall and the MOF subset of the CSD, sorted by count in the full CSD. Frequencies of occurrence are given in percentage.

	CSD overall			MOF subset		
Mo	6380	2.0	I (0.8), II (11.8), III (5.7), IV (14.3), V (20.0), VI (47.4)	535	1.0	II (4.9), III (1.5), IV (4.3), V (26.4), VI (63.0)
Sn	5855	1.8	I (0.1), II (19.6), III (0.1), IV (80.2), V (0.0), VI (0.0)	469	0.9	II (9.6), IV (90.4)
Rh	5555	1.7	I (45.9), II (14.6), III (39.2), IV (0.2), V (0.1)	160	0.3	I (7.5), II (85.0), III (7.5)
V	5356	1.6	I (0.9), II (3.8), III (13.4), IV (39.6), V (42.3), VI (0.0)	448	0.8	II (0.4), III (11.2), IV (45.3), V (42.9), VI (0.2)
Re	4684	1.4	I (30.5), II (4.7), III (12.9), IV (6.5), V (37.0), VI (1.8), VII (6.7)	18	0.0	I (22.2), II (5.6), III (44.4), IV (5.6), V (16.7), VII (5.6)
Ti	4437	1.4	I (0.0), II (3.1), III (13.0), IV (83.9)	77	0.1	II (1.3), III (11.7), IV (87.0)
Cr	4114	1.3	I (2.2), II (15.9), III (73.8), IV (1.7), V (2.2), VI (4.3)	329	0.6	I (0.6), II (10.0), III (87.5), V (0.3), VI (1.5)
Hg	3906	1.2	I (1.4), II (98.6)	681	1.2	I (2.6), II (97.4)
Ir	3622	1.1	I (21.6), II (3.0), III (73.7), IV (1.2), V (0.6)	20	0.0	I (15.0), III (75.0), IV (10.0)
Pb	3010	0.9	I (0.1), II (95.8), III (0.0), IV (4.1)	1067	1.9	II (99.0), IV (1.0)
Eu	2560	0.8	II (8.2), III (91.8)	899	1.6	II (3.1), III (96.9)
U	2757	0.8	II (0.4), III (9.4), IV (32.3), V (6.8), VI (51.0)	332	0.6	III (0.3), IV (12.3), V (1.5), VI (85.8)
W	2682	0.8	I (0.4), II (14.6), III (3.4), IV (17.2), V (14.5), VI (49.8)	276	0.5	IV (7.6), V (28.6), VI (63.8)
Dy	1816	0.6	II (0.8), III (99.2)	439	0.8	II (0.7), III (99.3)
Gd	1997	0.6	II (0.5), III (99.5)	643	1.2	II (0.3), III (99.7)
Sm	1881	0.6	II (15.6), III (84.4)	503	0.9	II (2.2), III (97.8)
La	1712	0.5	II (0.3), III (99.7)	576	1.0	III (100.0)
Nd	1748	0.5	II (0.4), III (99.6)	559	1.0	III (100.0)
Tb	1673	0.5	II (0.5), III (99.5), IV (0.1)	629	1.1	II (0.2), III (99.7), IV (0.2)
Zr	1693	0.5	II (3.5), III (4.5), IV (92.0), VI (0.1)	136	0.2	III (0.7), IV (98.5), VI (0.7)
Yb	1709	0.5	II (28.0), III (72.0)	238	0.4	II (2.5), III (97.5)
Er	1190	0.4	II (0.6), III (99.2), IV (0.2)	392	0.7	II (1.0), III (98.7), IV (0.3)
Bi	1424	0.4	II (1.4), III (92.3), V (6.2)	216	0.4	II (1.4), III (96.3), V (2.3)
Y	1429	0.4	I (0.1), II (0.1), III (99.8), IV (0.1)	215	0.4	III (100.0)
Ce	1293	0.4	I (0.1), II (0.6), III (75.5), IV (23.8)	318	0.6	II (0.3), III (89.6), IV (10.1)
Os	1334	0.4	I (0.5), II (46.0), III (14.8), IV (21.0), V (1.6), VI (15.1), VII (0.1), VIII (1.0)	22	0.0	II (81.8), III (9.1), IV (9.1)
In	1366	0.4	I (6.1), II (2.7), III (91.1)	269	0.5	II (0.4), III (99.6)
Ga	1063	0.3	I (10.1), II (7.1), III (82.8)	73	0.1	I (1.4), II (2.7), III (95.9)
Nb	860	0.3	I (3.4), II (2.7), III (10.8), IV (20.2), V (62.9)	56	0.1	III (3.6), IV (73.2), V (23.2)
Pr	1020	0.3	II (0.3), III (99.5), IV (0.2)	421	0.8	III (99.8), IV (0.2)
Mg	994	0.3	I (2.0), II (98.0)	213	0.4	II (100.0)
Li	568	0.2	I (99.5), II (0.2), III (0.2), IV (0.2)	133	0.2	I (98.5), II (0.8), IV (0.8)
Na	779	0.2	I (99.9), III (0.1)	337	0.6	I (100.0)
Ho	676	0.2	II (0.7), III (99.3)	235	0.4	II (0.4), III (99.6)
K	512	0.2	I (100.0)	182	0.3	I (100.0)
Ta	758	0.2	I (1.7), II (1.3), III (6.7), IV (12.1), V (78.1)	5	0.0	II (20.0), V (80.0)
Ca	677	0.2	I (0.1), II (99.9)	290	0.5	II (100.0)
Tl	762	0.2	I (63.1), II (1.3), III (35.6)	156	0.3	I (76.9), II (0.6), III (22.4)
Ba	466	0.1	II (100.0)	219	0.4	II (100.0)

Continued on next page

Table 23: Distribution of metals and oxidation states in the CSD overall and the MOF subset of the CSD, sorted by count in the full CSD. Frequencies of occurrence are given in percentage.

	CSD overall			MOF subset		
Np	190	0.1	II (0.5), III (3.2), IV (22.6), V (46.8), VI (26.3), VII (0.5)	48	0.1	IV (18.8), V (62.5), VI (18.8)
Sr	370	0.1	II (100.0)	192	0.3	II (100.0)
Sc	431	0.1	I (0.5), II (1.2), III (98.4)	44	0.1	III (100.0)
Tm	313	0.1	II (12.1), III (87.9)	71	0.1	III (100.0)
Lu	449	0.1	II (0.2), III (99.8)	66	0.1	III (100.0)
Hf	462	0.1	II (1.7), III (2.4), IV (95.9)	26	0.0	III (3.8), IV (96.2)
Th	435	0.1	II (0.5), III (2.1), IV (97.2), VI (0.2)	38	0.1	III (2.6), IV (94.7), VI (2.6)
Pu	127	0.0	III (26.0), IV (48.0), V (4.7), VI (21.3)	19	0.0	III (26.3), IV (31.6), V (10.5), VI (31.6)
Rb	78	0.0	I (98.7), II (1.3)	31	0.1	I (100.0)
Cs	77	0.0	I (98.7), II (1.3)	36	0.1	I (100.0)
Be	49	0.0	II (100.0)	2	0.0	II (100.0)
Am	26	0.0	III (80.8), V (7.7), VI (11.5)	5	0.0	III (60.0), V (20.0), VI (20.0)

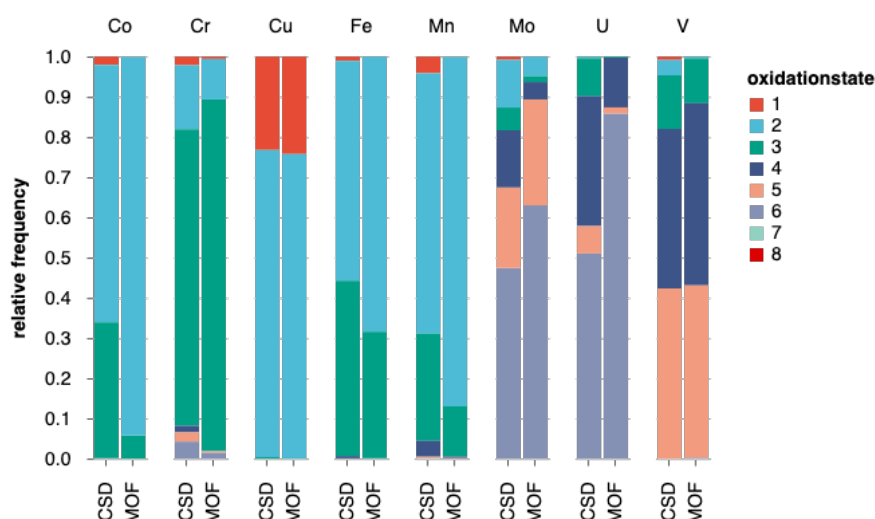


Figure 117: Oxidation state frequencies. Relative frequencies of oxidation states for transition metals that are common for MOFs and that occur in more than one oxidation state (> 0.05 relative frequency for minor oxidation states).

Figure 117 compares the relative frequency of oxidation states for common transition metals in the MOF subset (94 716 structures in total, from the release of May 2019) with the full CSD dataset of more than one million structures. From Figure 117, one can observe that the distribution of oxidation states in the MOF subset is not drastically different from the overall distribution, which justifies our focus on MOFs.

Also, this analysis showcases that turning the assignment of oxidation states into a classification problem with strong priors is possible. Each metal only occurs in a small number of oxidation states, e.g., for copper only the oxidation states +I and +II are relevant for MOF chemistry. In some cases, like for alkali or alkaline earth metals, only one oxidation state is reported, and the metal will immediately be associated with only that oxidation state. We also used this exploratory analysis to find potentially wrong assignments in the CSD by manually analyzing all the entries with rare, sometimes nonphysical, oxidation states. These cases were excluded from training, as discussed in the next section (cf. Section D.1.5).

D.1.5 Structures excluded from training set

Mixed valence structures

Our text-parsing pipeline was not built to resolve the chemical names, so we cannot assign the oxidation state to a particular metal site. Hence, we excluded cases where the same metal occurred in two different oxidation states in the same structure (mixed-valence compounds) from the training set. Figure 118 shows the number of such cases for each metal. Note that our model can deal with type 1 mixed-valence⁵²⁵ compounds as we treat each metal site separately.

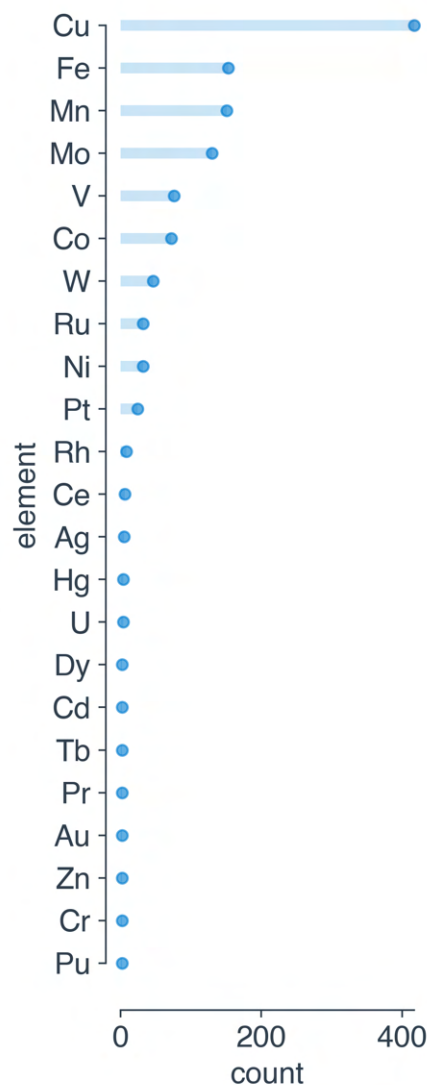


Figure 118: Type 1 mixed-valence compounds. Number of cases in which the element occurs multiple times in different oxidation states as a function of the metal. Most often, these are MOFs containing both Cu(I) and Cu(II) metal nodes in the same structures.

Uncommon oxidation states

In our model, we do not constrain *a priori* the possible oxidation states depending on the metal. This means selecting a general range of oxidation states for all the metals that the model can predict.

In the choice of this range, we excluded classes (i.e., oxidation states) with less than ten examples, as no reliable cross-validated training can be performed for those cases. We found that this is only the case for the oxidation state seven, for which there is only one example in the CSD, namely a structure (BOJSUO) with Re(VII) centers in which Re_2O_7 units are linked by solvent (dioxan) molecules. For this reason, we concluded that it is reasonable to limit our model to the oxidation states one to six as also the oxidation states 0 and lower as well as VIII and higher⁵²⁶ are not relevant for MOFs and not well represented in the CSD.

Cases excluded after exploratory data analysis

During EDA of the MOF subset of the CSD we investigated all the oxidation state/metal combinations with less than 20 examples by comparing the entries in the CSD with the assignment in the paper to confirm the reliability of the chemical names in the CSD, and to also ensure that our training set does not contain nonphysical oxidation state assignments. As a result of this process, we excluded the structures in Table 24 from our training and test set as we found disagreements between the oxidation state in the name in CSD and the assignment given in the paper or nonphysical oxidation states like Li(IV). This filtering highlights the power exploratory analysis on big data can have, as we could already capture some mistakes in the underlying data source without major efforts and chemical analysis.

Table 24: Potential errors in the CSD found by means of EDA. Materials excluded after EDA due to discrepancy with oxidation state in original paper or nonphysical oxidation state.

CSD code	reference	metal centre and assignment in the chemical name in the CSD	reason for exclusion
BIZLOO ⁵²⁷		Zr(VI)	non-physical oxidation state
AFEHOL ⁵²⁸		Mn(I)	rare oxidation state without experimental evidence
SULPMS ⁵²⁹		Ag(II)	Ag(I) in original paper
ADASUW ⁵³⁰		Ag(II)	Ag(I) in original paper
FENXAY ⁵³¹		Ag(II)	more likely Ag(I)
KAWCES ⁵³²		Ag(II)	more likely Ag(I)
AMUTEI ⁵³³		Ag(II)	more likely Ag(I)
MAJLOZ ⁵³⁴		Ag(II)	more likely Ag(I)
WAZKOZ ⁵³⁵		Ag(II)	more likely Ag(I)
LAVYIT ⁵³⁶		Ag(II)	Ag(I) in original paper
EQEHUE ⁵³⁷		Ag(II)	Ag(I) in the original paper
VOMMUH ⁵³⁸		Ag(II)	Ag(I) in original paper
MITSIS ⁵³⁹		Cd(I)	Cd(II) in paper
WAQFAY ⁵⁴⁰		Cd(III)	Cd(II) in paper
ARADEE ⁵⁴¹		Cd(III)	Cd(II) in paper
MAVLED ⁵⁴²		Cd(III)	Cd(II) in paper
ZEQROE ⁵⁴³		Cd(III)	no evidence for unusual oxidation state
RAWFAZ ⁵⁴⁴		Ce(II)	Ce(III) in the original paper
XEDJUN ⁵⁴⁵		V(II)	V(IV) in paper
TEJFOG ⁵⁴⁶		Hg(I)	Hg(II) in paper
ZEJWOD ⁵⁴⁷		Gd(II)	Gd(III) in paper
KEPGES ⁵⁴⁸		Gd(II)	Gd(III) in paper

Continued on next page

Table 24: Potential errors in the CSD found by means of EDA. Materials excluded after EDA due to discrepancy with oxidation state in original paper or nonphysical oxidation state.

CSD code	reference	metal centre and assignment in the chemical name in the CSD	reason for exclusion
BUHVOP ⁵⁴⁹		Er(IV)	Er(IV) not known
WAQKIK ⁵⁵⁰		Dy(II)	Dy(III) in paper
ZEJXEU ⁵⁴⁷		Dy(II)	Dy(III) in paper
KEXZUI ⁵⁵¹		Tl(II)	Tl(I) in original reference
REYDUX ⁵⁵²		Li(II)	non-physical oxidation state
ZARBEZ ⁵⁵³		Li(IV)	non-physical oxidation state
NENNOK ⁵⁵⁴		Ni(I)	Ni(II) in the original paper, though Ni(I) in title
WOQKET ⁵⁵⁵		Co(I)	Co(II) in the original paper
VEYJOB ⁵⁵⁶		V(VI)	nonphysical oxidation state, V(V) in paper
MITFON ⁵⁵⁷		Th(VI)	nonphysical, Th(IV) in original paper
IJATUI		Eu(II)	Eu(III) in the original paper ⁵⁵⁸
ZEJWIX		Er(II)	Er(III) in the original paper ⁵⁴⁷
COFYOM		Er(II)	Er(III) in the original paper ⁵⁵⁹
KEPGIW		Er(II)	Er(III) in the original paper ⁵⁴⁸
VUNQUS		Sm(II)	Sm(III) in the original paper ⁵⁶⁰
ZEJXAQ		Ho(II)	Ho(III) in the original paper ⁵⁴⁷
MAPGUI		Yb(II)	Yb(III) in the original paper ⁵⁶¹

LANTHANIDES IN OXIDATION STATE +II IN THE MOF SUBSET

Given that lanthanides in oxidation state +II are harder to access in typical MOF synthesis, we list below the relevant CSD reference codes in the MOF subset.

- *Europium*: MAKWUS,⁵⁶² EDINAG,⁵⁶³ IJATUI (see Table 24), LEQRUU,⁵⁶⁴ MAVRUX,⁵⁶⁵ WIDREH,⁵⁶⁶ LIBDAB,⁵⁶⁷ YAVXEY,⁵⁶⁸ LIBCUU,⁵⁶⁷ REJXEJ,⁵⁶⁴ LEQTIK,⁵⁶⁴ YAWMIT,⁵⁶⁹ ZEQWAT,⁵⁷⁰ IQEZAF,⁵⁷¹ WUJKUI,⁵⁷² MOXQIB,⁵⁷³ TAKDIU,⁵⁷⁴ PIKJUR,⁵⁷⁵ AGUROK,⁵⁷⁴ EHAZES,⁵⁷⁶ ULUCUZ,⁵⁷⁷ MAKWIG,⁵⁶² YAVXAU,⁵⁶⁸ MAKWOM,⁵⁶² QALNEW,⁵⁶³ JEZZOG,⁵⁷⁸ NOQYOH⁵⁷⁹
- *Gadolinium*: ZEJWOD (see Table 24), KEPGES (see Table 24)
- *Erbium*: ZEJWIX (see Table 24), WUKNEV,⁵⁸⁰ COFYOM (see Table 24), KEPGIW (see Table 24)
- *Samarium*: JODXOS,⁵⁸¹ UCEGAK,⁵⁸² UCEGOY,⁵⁸² UCEGEO,⁵⁸² UCEFUD,⁵⁸² UCEHAL,⁵⁸² UCEGUE,⁵⁸² VUNQUS (see Table 24), GONGUL,⁵⁸³ WEBHUG,⁵⁸⁴ TABHIP⁵⁸⁵
- *Holmium*: ZEJXAQ (see Table 24)
- *Ytterbium*: IQEYUY,⁵⁷¹ LIBDEF,⁵⁶⁷ XULTAZ,⁵⁸⁶ UGOTOZ,⁵⁸⁷ MAPGUI (see Table 24), UGOTIT⁵⁸⁷
- *Cerium*: RAWFAZ (see Table 24)
- *Terbium*: FIFVUN⁵⁸⁸
- *Dysprosium*: WAQKIK (see Table 24), ZEJXEU (see Table 24), VAYDUW (mixed valence)⁵⁸⁹

D.1.6 Diverse set selection

In practice, one does not want to use all available data for training due to constraints in computational time and memory (and the risk of overfitting for a highly redundant dataset). Following Pauling’s parsimony principle, and the fact that entries in the CSD can be the same MOF with a slightly varied linker, we wanted to distill the data set to a non-redundant set of chemical environments. Note that if the local chemical environments around the metal centers were exactly the same in fingerprint space, we would, anyway, automatically exclude them. To perform the diverse set selection, we use the *apricot* Python package³⁴⁴ which implements submodular selection using facility location functions and several algorithmic optimizations. Those facility location functions take the form

$$f(X) = \sum_{y \in Y} \max_{x \in X} \phi(x, y), \quad (21)$$

and select a subset X from the data set Y , with ϕ as a similarity measure, for which we used the Euclidean distance.

Due to memory constraints (as evaluating eq. 21 involves computing the square distance matrix), we applied the algorithm on random batches of the full data set that we could fit into memory. To address the fact that some elements are more frequent in the CSD than others, we used a random set of 5000 structures for elements that occur more frequently than 5000 times (random subsampling) before we applied the sub-modular selection algorithm.

D.1.7 Representative environments

To better understand the distribution of structures in the chemical space spanned by our feature vectors, we used the *mmd-critic* framework.⁵⁹⁰ This technique selects *prototypes*, which are examples that are representative of the distribution of structures in the chemical space spanned by the features we used for our models. In addition, it also selects *critics*, which are also real data points that represent parts that are not well captured by the *prototypes*. Those critics lead to an increase in interpretability compared to other techniques such as *k*-medoid clustering. Note that this analysis is fully unsupervised, i.e., we do not consider the oxidation states but only the feature matrix.

The technique is built around the maximum mean discrepancy (MMD) between two distributions P and Q over the function space \mathcal{F} :

$$\text{MMD}(\mathcal{F}, P, Q) = \sup_{f \in \mathcal{F}} (E_{X \sim P} [f(X)] - E_{Y \sim Q} [f(Y)]). \quad (22)$$

The prototypes S are a subset of X that minimize the MMD between S and X . The critics, on the other hand, are selected such that the difference between the dataset X and the set S is maximal. These optimization problems can be written with kernel functions (if \mathcal{F} is a reproducing kernel Hilbert space), and the problem reduces to minimization and maximization of a witness function, respectively.

We chose the initial Gaussian kernel width ($\gamma = 0.085$) for this analysis based on the median heuristic⁵⁹¹ and then measured the one-nearest-neighbor classification performance (trained on 500 prototypes) for different kernel widths between the 0.1 and 0.9 quartiles of 5000 random distances between the data points in the feature set. The code for this analysis is available in our fork of the *mmd-critic* code (<https://github.com/kjappelbaum/MMD-critic.git>).

Figure 119 shows ten prototype environments and Figure 120 summarizes ten critics environments (the images were created using Mercury). Common characteristics

the critics show are unusual coordination geometries (e.g., ten-fold coordinated thorium center in BIFPOY) or uncommon compound classes (e.g., organoindium, organotin, organogermanium or coordination polymers formed by Cp^* ligands) that are rare in the MOF subset of the CSD. The prototypes, on the other hand, showcase more common metals in prototypical environments.

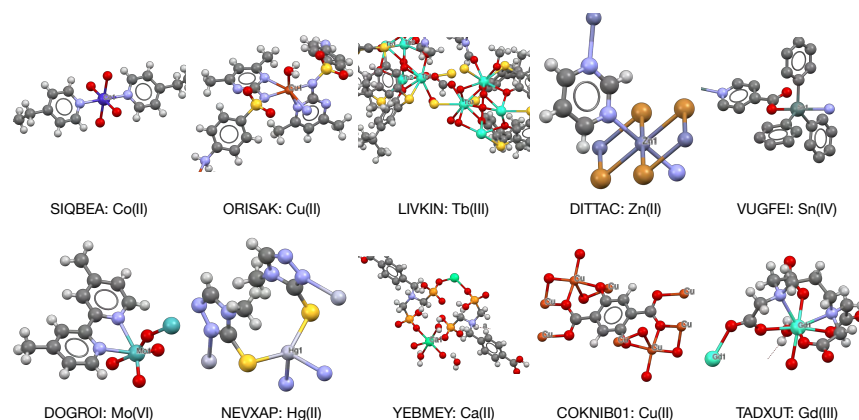


Figure 119: Prototypes. Overview over ten prototypes, i.e., structures that are representative of the dataset.

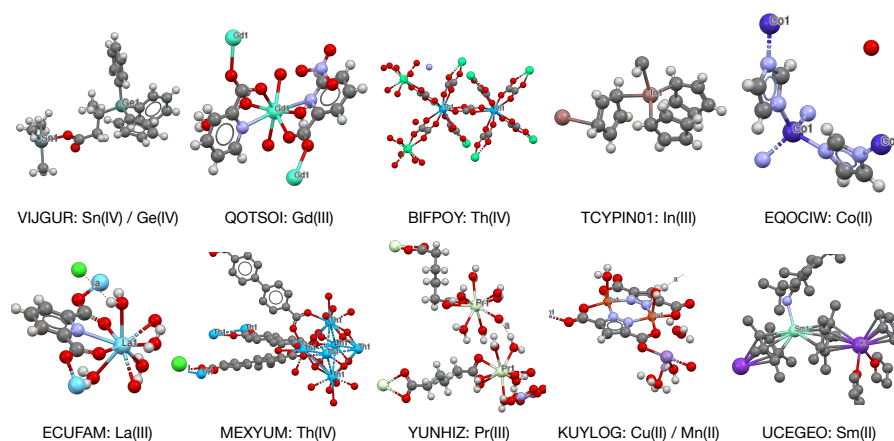


Figure 120: Critics. Overview over ten critics, i.e., structures that are not well represented by the dataset.

D.2 MODEL DESCRIPTION AND ANALYSIS

D.2.1 Model architecture choice

Given that each element only exists in a limited number of oxidation states that is usually smaller than the total number of possible oxidation states for all elements, one could also imagine training a separate classifier for each model. We decided not to use this architecture to leverage the patterns that exist in chemistry and, in this way, also to be able to attempt predictions for elements for which little data is available but for which other elements in the training set exist that show similar chemistry (e.g., in the same column).

Also, we chose not to hard-code any rules, such as fixing oxidation state +I for alkali metals, to help the model learn patterns across the periodic table—and also to be able to verify that the model indeed learns such patterns across the periodic table. In any case, such heuristics can still be applied on top of the predictions of our model.

Similarly, we do not take the overall stoichiometry of the compound into account to ensure that the model is robust w.r.t errors in the protonation or the number of charge-compensating counter-ions. Latter is important for a real-world application of our model, as in setting up molecular simulations, one typically tries to deduce the correct protonation from the oxidation state of the metal center.

D.2.2 Class imbalance

To avoid poor generalization due to imbalanced class distributions, we also tried to employ oversampling techniques (like SMOTE and ADASYN)⁵⁹² but did not observe an increase in predictive performance that would justify the increase in computational complexity and the risk of overfitting. We found that we can achieve good performance by subsampling the most frequent classes and a diverse set selection (cf. section D.1.6). We envision that a more promising approach for future work is to sample structures with oxidation states that are uncommon for MOFs from other parts of chemistry (i.e., the remaining parts of the CSD). The results of some initial attempts are reported in section D.8.

D.2.3 Metrics

To evaluate the performance of our models, we considered different metrics, which are defined as follows (for the binary classification case), using true positive (TP), true negative (TN), false positive (FP), false negative (FN):

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (23)$$

$$\text{precision} = \frac{TP}{TP + FP} \quad (24)$$

$$\text{recall} = \frac{TP}{TP + FN} \quad (25)$$

$$\text{balanced accuracy} = \frac{1}{2} \left(\frac{TP}{TP + FP} + \frac{TN}{TN + FN} \right) \quad (26)$$

$$F_1 = \frac{2TP}{2TP + FP + FN} \quad (27)$$

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (28)$$

One metric that is often used is the AUC. This metric measures the area under a curve of the FP rate (1-specificity) plotted against the TP rate (sensitivity). An ideal classifier has a high specificity combined with high sensitivity, hence an AUC close to one.

For multi-class problems, one can directly use Cohen's κ :⁵⁹³

$$\kappa = 1 - \frac{1 - p_0}{1 - p_e}, \quad (29)$$

where p_0 is the observed accuracy and p_e is the accuracy a random classifier would achieve.

To extend the other metrics to the multi-class setting, one can calculate micro and macro averages, respectively. In macro averaging, the metric is calculated independently for each class and then averaged, whereas in micro averaging the individual TP, TN, FP, FN for each class are summed up.

Note that we calculate the metrics based on the individual metal sites (and not for the whole structures).

D.2.4 Training and model selection

For initial model selection on small feature sets, we used 10-fold stratified cross-validation to estimate the prediction errors as it was shown^{594,595} to provide a good balance between variance and bias at a reasonable computational cost. For larger training sets (> 10000) we fell back to a holdout test set. We used iterative stratification (on the oxidation states and the atomic number) to ensure that the training and test set fold class proportions are approximately equal.³⁴³ Note that we also perform diverse set selection as described in Section D.1.6.

For model selection, we evaluated ensembles of different classifiers, with both hard and soft voting, and optimized the hyperparameters for each base estimator using a mixed search strategy (to avoid a possible bias by using only one strategy) using hyperopt⁵⁹⁶ (using at most 500 evaluations for a maximum of 6 min using 80 % tree-Parzen estimator⁵⁹⁷ because of its high efficiency,⁵⁹⁸ 10 % random search²⁶⁷ and 10 % annealing) on the default search spaces in our fork (<https://github.com/kjappelbaum/hyperopt-sklearn>) of the hyperopt-sklearn⁵⁹⁹ package using a validation holdout set of 30 %.

We aimed to combine base estimators with different hypothesis spaces to make the model more robust and also to get an uncertainty estimate. A representative set of hyperparameters is shown in Table 25.

Table 25: Summary of representative model hyperparameters.

parameter	value
k-nearest neighbors (knn)	
leaf size	30
metric	manhattan
neighbors	5
weights	distance
extra trees (et)	
criterion	gini
max depth	None
max features	0.99662
max leaf nodes	None
min impurity decrease	0.0
min impurity split	None
min samples leaf	1
min samples split	2
min weight fraction leaf	0.0
estimators	64
gradient boosting (gb)	
criterion	friedman mse
learning rate	0.57012
loss	deviance
max depth	None
max features	0.7464

Continued on next page

Table 25: Summary of representative model hyperparameters.

parameter	value
max leaf nodes	None
min impurity decrease	0.0
min impurity split	None
min samples leaf	1
min samples split	2
min weight fraction leaf	0.0
estimators	23
n iter no change	None
subsample	0.58778
tol	0.0001
stochastic gradient descent (sgd)	
α	5.954895e-06
average	False
early stopping	False
ϵ	0.1
η_0	0.00116898
fit intercept	True
l_1 ratio	0.114230
learning rate	optimal
loss	log
max iter	148296682
penalty	l1
power t	0.34923
tol	2.92064e-05

Due to our need to use pre-trained calibrated base estimators in the final voting classifier, we implemented our custom `VotingClassifier` class based on the `sklearn` implementation, which is available in our `learnmofox` Python package. This class is needed to run or retrain the models.

D.2.5 Global and class performance metrics

The performance metrics listed below were calculated using the `pycm` package.⁶⁰⁰ In Table 26 we summarize some global statics. In Table 27 we show the class statistics and in Table 28 we show the confusion matrix.

Table 26: Global performance metrics.

metric	value
accuracy	0.997 ± 0.003
F_1 macro	0.98
F_1 micro	0.99
κ	0.987 ± 0.0006
$\kappa_{\text{no prevalence}}$ ⁶⁰¹	0.98
relative classifier information ⁶⁰²	0.96
precision macro	0.98
precision micro	0.99

D.2.6 Bootstrapped performance estimates

To evaluate the performance of our classifier, we used the bootstrap technique to calculate statistics for several classification metrics. Confidence intervals were calculated using the percentile method described by Efron,⁶⁰⁵ using the 2.5th and 97.5th percentiles of the distribution of metrics for the bootstraps.

Table 27: Summary of class statistics. Decimal places for values > 0.995 are cut and not rounded.

metric	I	II	III	IV	V	VI
accuracy	0.99	0.99	0.99	0.99	0.99	0.99
adjusted F score	0.99	0.99	0.99	0.99	0.96	0.99
AUC	0.99	0.99	0.99	0.99	0.95	0.99
AUPR	0.99	0.99	0.99	0.97	0.93	0.99
precision	0.99	0.99	0.99	0.96	0.96	0.99
Matthews correlation coefficient ⁶⁰³	0.99	0.99	0.98	0.97	0.93	0.99
Gini index ⁶⁰⁴	0.99	0.99	0.98	0.98	0.91	0.99

Table 28: Confusion matrix. Numbers in each cell show the number of classified metal centers for each case. In the ideal error-free case, the matrix would only have entries on the diagonal.

		prediction					
ground truth		I	II	III	IV	V	VI
	I	12604	34	3	0	0	4
	II	84	34399	53	12	0	0
	III	18	184	9555	12	0	0
	IV	0	8	8	2302	24	0
	V	0	0	0	54	614	10
	VI	0	4	0	8	0	2724

D.2.7 Comparison with baseline (floor) metrics

As baselines for our metrics, we used two dummy classifiers that arrive at the prediction by guessing uniformly at random (drawing from a uniform distribution of oxidation states) or guessing at random while respecting the training set class distribution (stratified). The results shown in Table 29 show that our model performs substantially better than random guessing. The models for these analyses were trained on 47951 examples and tested on 42463 disjoint test cases.

Table 29: Baselines. Baseline metrics, derived using dummy classifiers (for all metals, and not only Cu, as shown in the main text in Figure 2). Median effect sizes (all $p < 0.01$) relative to the *best* performing baseline classifier.

metric	stratified random guessing	uniform random guessing	majority	our model	minimum effect size
accuracy	0.40	0.17	0.57	0.98	0.41
balanced accuracy	0.17	0.17	0.17	0.96	0.80
F_1 micro	0.40	0.17	0.57	0.98	0.41
F_1 macro	0.17	0.12	0.12	0.96	0.80
precision	0.40	0.17	0.57	0.98	0.41
recall	0.40	0.17	0.57	0.98	0.41

D.2.8 Randomization tests to estimate the significance of the classifier scores

To further ensure that the classifier learned a real structure in the data, we also performed permutation tests in which the classification procedure is repeated 200 times with permuted labels.^{606,607} If the classifier learned a real structure in the data,

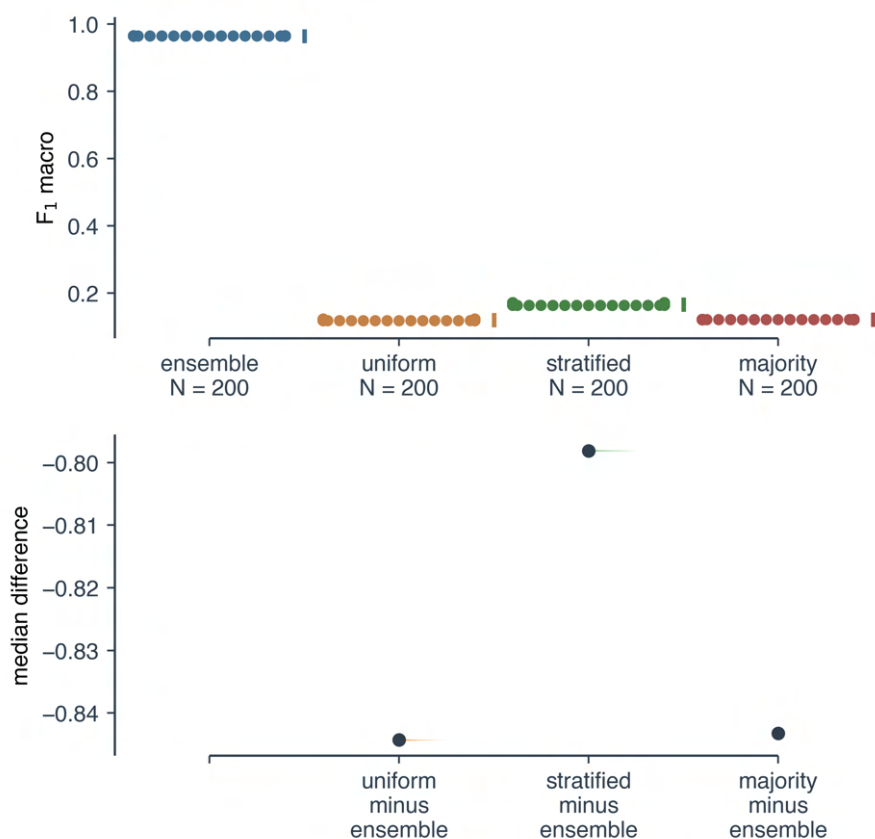


Figure 121: Difference between baselines and our model. Median effect size of the F_1 macro score using 200 bootstraps for the test set and 10000 bootstraps for the effect size.

it should perform significantly better than the straw models.⁶⁰⁸ Here, we used 10-fold cross-validation to measure each model's empirical error.

For efficiency reasons (poor convergence without structure in the data), we performed this test on a subset of the data (100 sites), which we selected using submodular selection. For this reason, the score for the model trained on non-permuted data is also lower.

Figure 122 shows a histogram for the balanced accuracies on a holdout test set and indicates that our model performs significantly ($p = 0.005$) better than the straw models that contain no chemical information.

D.2.9 Reliability diagram

Reliability diagrams plot the actual probabilities against the predicted probabilities and are useful to assess whether the probabilities which a model produces can be interpreted as confidence scores. An ideal classifier (reliable model) would fall onto the diagonal, but one usually finds distorted curves as, for example, ensemble classifiers tend to predict less frequently very high and very low probabilities.⁶⁰⁹ As these diagrams are constructed using binning operations, some fluctuations arise due to varying bin sizes or bin means. To take this into account, we use a revised consistency bars technique that provides consistency bars that are the .5 and .95 quantiles derived from bootstrapping.⁶¹⁰

From this analysis (on 10000 holdout examples with 1000 bootstraps) we find

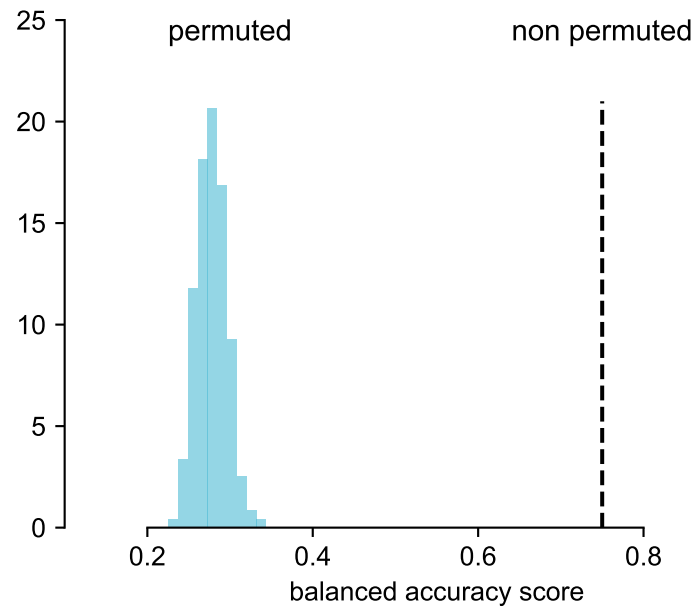


Figure 122: Permutation experiment to determine the significance of classification results. Histogram of balanced accuracy scores (ten-fold cross-validated) for classification with permuted labels.

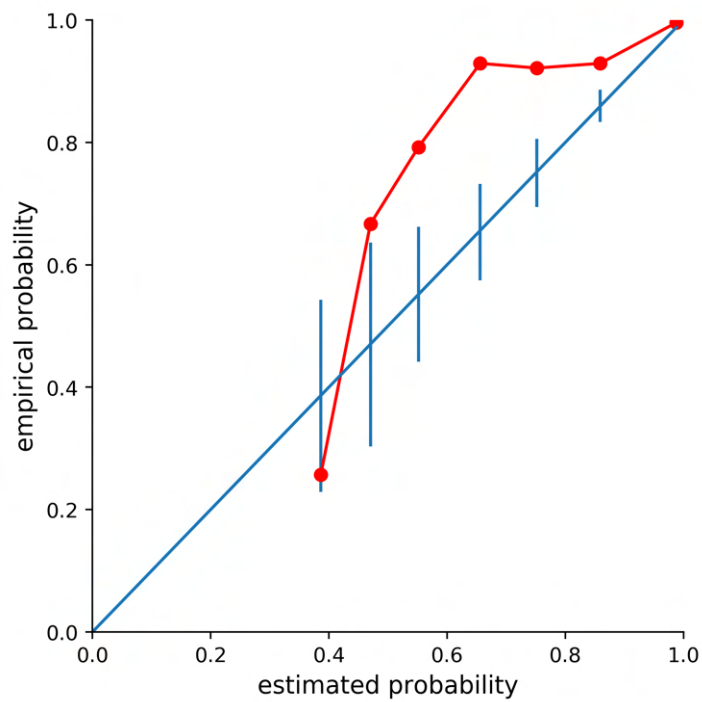


Figure 123: Reliability diagram with consistency bars.

that our classifier could even increase its confidence in the predictions with more than 50 % confidence and predict probabilities close to one more often (cf. Figure 123).

D.2.10 Learning curves

We recorded learning curves to measure the predictive performance's dependence on the training set size. To do so, we ran our training pipeline (including probability calibration) for different training set sizes and recorded the metrics on the fixed-size holdout set of 63790 examples. Note that for doing so, we left the hyperparameters fixed as the optimal ones we found for the largest training set size. In Figure 124, we also provide, as visual aides, the baseline metrics (cf. Figure 29). One can observe that even with the smallest training sets, we perform better than the baseline. Using tens of thousands of data points, all metrics exceed a score of .96.

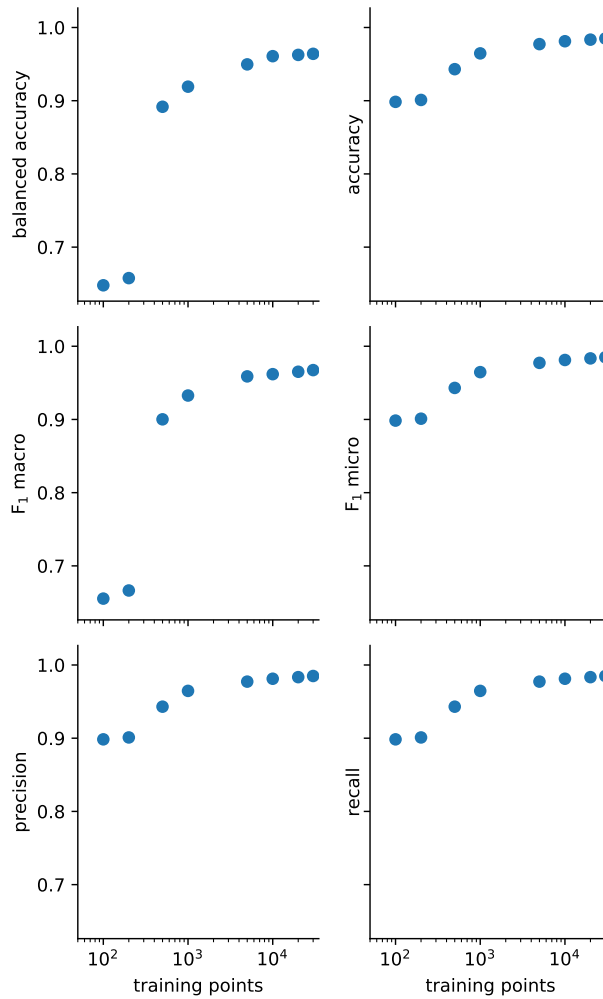


Figure 124: Learning curves. Metrics as a function of the number of training and validation points.

D.2.11 Ensemble property analysis

Even though it is well-established that ensemble models improve the stability and predictive performance (for uncorrelated base estimators),⁶¹¹ we compared the performance of the optimized base estimators with the final ensemble model on the test set.

Uncertainty estimate via voting agreement

To quantify the quality of the uncertainty estimate, we calculated bootstrapped ($n = 5000$) effect sizes which are summarized in Table 30 and which indicate that the ensembling approach gives us a significant and relevant estimate about uncertain predictions.⁶¹² This is also observable in Figure 125, where we plot the number of disagreeing base estimators for true and correct predictions and the bootstrapped mean effect size.

Table 30: Uncertainty estimate via voting agreement. Bootstrapped effect sizes for the number of base estimators that disagree with the final prediction of the ensemble model.

measure	effect size [95 % confidence interval]	statistical test
mean difference	0.89 [0.81,0.96]	t -test: $p = 0.0$
median difference	1.0 [1.0,1.0]	Kruskal-Wallis: $p = 0.0$
Hedges' g	3.6 [3.3, 3.9]	t -test: $p = 0.0$

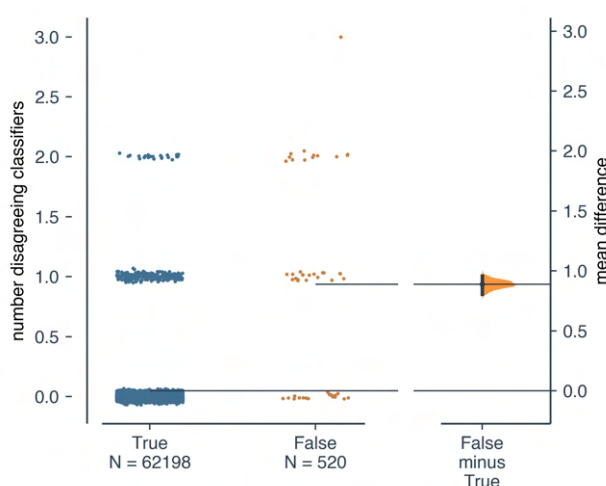


Figure 125: Uncertainty estimate via voting agreement. Number of base estimators that disagree with the final prediction for correct and incorrect predictions (we subsampled and added noise to the data for the swarmplot for a clearer representation) and the bootstrap mean effect size estimate.

Note that this uncertainty estimate is not perfect. Even if all models agree, the prediction can still be wrong.

D.3 FEATURIZATION

D.3.1 Detailed descriptions of features considered in this work

Obviously, a model that only considers the local environment will not be able to predict non-integer oxidation states as they are present in some (type-III) mixed-valence⁵²⁵ structures like the Creutz-Taube ion,^{613,614} where two metal centers with equivalent local environments exhibit inner-sphere electron transfers. This could be addressed by using non-integer oxidation states as class labels, possibly with the addition of global features or a message-passing model architecture.

Metal center

To encode the metal center, we considered the row, column, and atomic number as well as the number of valence electrons (also encoded as the minimal distance to an 18-electron shell) and the number of unfilled s, p, and d orbitals (all properties for the ground state atom). We retrieved those properties using `Magpie`⁶¹⁵ and `pymatgen`.¹⁷⁴

Chemistry

To encode the local chemistry, we used an extended set of the descriptors proposed by Ward *et al.*³³¹ We calculated this descriptor vector using `matminer` and included signed and maximum, minimum as well as average local differences (signed and unsigned) of the Mendeleev number, row, columns, electronegativity, number of s valence electrons, number of p valence electrons, number of d valence electrons, number of f valence electrons, number of valence electrons, number of unfilled s electrons, number of unfilled p electrons, number of unfilled d electrons, number of unfilled f electrons, number of unfilled electrons as well as the ground state band-gap (indicates how metallic an element is).

Geometry

To encode the local coordination geometry, we used the `CrystalNNFunction` implemented in `matminer` which calculates Steinhardt bond orientational order parameters⁶¹⁶ and the similarity to different coordination environments using the approach described by Zimmermann *et al.*^{165,329} We also considered adding symmetry functions suggested by Behler,⁶¹⁷ which were also calculated using the `matminer` package ($\eta_{G_2} = [0.05, 5, 20, 80]$, $\eta_{G_4} = 0.005$, $\zeta_{G_4} = [1, 4]$, cutoff 6.5 Å) but did not use them in the final model, following the minimal descriptor length principle.

Revised autocorrelation functions (RACs)

RACs are the extended and tailored version of autocorrelations (ACs)⁶¹⁸ descriptors for transition metal chemistry,¹⁶³ which were recently adapted to MOF chemistry.¹³² RACs use a graph representation of a molecule/crystal to correlate atomic properties between atoms separated by a certain number of bonds. The nodes of this graph representation are atoms labeled with atom types, and two nodes are connected when there is a chemical bond between the corresponding atoms. The correlations used for RACs in this study are the product of or difference between atomic properties, which are computed using these equations:

$$\sum_{\text{start}}^{\text{start}} \sum_{\text{scope}}^{\text{scope}} P_d^{\text{prod}} = \sum_i^{\text{start}} \sum_j^{\text{scope}} (P_i P_j) \delta(d_{i,j}, d) \quad (30)$$

$$\sum_{\text{start}}^{\text{start}} \sum_{\text{scope}}^{\text{scope}} P_d^{\text{diff}} = \sum_i^{\text{start}} \sum_j^{\text{scope}} (P_i - P_j) \delta(d_{i,j}, d). \quad (31)$$

In these equations, atomic property P of atom i selected from the start atom list is correlated with atom j selected from the scope atom list if they are separated by d number of bonds. $d_{i,j}$ is the shortest path between the two atoms on the graph.

In this work, for each metal center of a MOF structure, we compute RACs with the start atom list being only the metal center and the scope atom list including all the atoms of the structure. RACs were computed up to the maximum depth of

three bonds using five heuristic atomic properties, namely atom identity (I), connectivity (T), Pauling electronegativity (χ), covalent radius (S), and nuclear charge (Z). To construct the graph for each metal center, we compute the adjacency matrix based on the periodic pairwise distances of atoms in the crystal. We assign a bond between two atoms when their pairwise distance multiplied by a tuning factor is below the sum of the covalent radii of the two atoms. The tuning factor is set to be 0.9 except for metal-organic pairs, which we tune depending on the atom types. No metal-metal bond is allowed in our graph representation. The code to compute these features is available in the `molSimplify` package.⁶¹⁹

Using only RACs, we could not achieve satisfactory performance (cf. Table 31), and we did not add them to our final vector vectors due to the minimum descriptor length principle and the fact that there are technical problems with calculating them for disordered structures.

FEATURE LIST USED IN THE FINAL MODEL The set of features we used for the model with which we obtained the results shown in the main text is listed below. The name in quotes is the name which is the name which we used in our code and data files.

Chemistry

- “local difference in MendeleevNumber”: Absolute local difference in the atomic numbers
- “local difference in Column”: Absolute local difference in the column (group) number
- “local difference in Row”: Absolute local difference in the row number (period)
- “local difference in Electronegativity”: Absolute local difference in electronegativity
- “local difference in NsValence”: Absolute local difference in the number of s electrons in the ground state atom
- “local difference in NpValence”: Absolute local difference in the number of p electrons in the ground state atom
- “local difference in NdValence”: Absolute local difference in the number of d electrons in the ground state atom
- “local difference in NfValence”: Absolute local difference in the number of f electrons in the ground state atom
- “local difference in NValence”: Absolute local difference in the number of valence electrons in the ground state atom
- “local difference in NsUnfilled”: Absolute local difference in the number of unfilled s electrons in the ground state atom
- “local difference in NpUnfilled”: Absolute local difference in the number of unfilled p electrons in the ground state atom
- “local difference in NdUnfilled”: Absolute local difference in the number of unfilled d electrons in the ground state atom
- “local difference in NfUnfilled”: Absolute local difference in the number of unfilled f electrons in the ground state atom
- “local difference in NUUnfilled”: Absolute local difference in the number of unfilled electrons in the ground state atom

- “local difference in GSbandgap”: Absolute local difference in the ground state bandgap of the bulk material
- “local signed difference in MendeleevNumber”: Signed local difference in the atomic numbers
- “local signed difference in Column”: Signed local difference in the column (group) number
- “local signed difference in Row”: Signed local difference in the row number (period)
- “local signed difference in Electronegativity”: Signed local difference in electronegativity
- “local signed difference in NsValence”: Signed local difference in the number of s electrons in the ground state atom
- “local signed difference in NpValence”: Signed local difference in the number of p electrons in the ground state atom
- “local signed difference in NdValence”: Signed local difference in the number of d electrons in the ground state atom
- “local signed difference in NfValence”: Signed local difference in the number of f electrons in the ground state atom
- “local signed difference in NValence”: Signed local difference in the number of valence electrons in the ground state atom
- “local signed difference in NsUnfilled”: Signed local difference in the number of unfilled s electrons in the ground state atom
- “local signed difference in NpUnfilled”: Signed local difference in the number of unfilled p electrons in the ground state atom
- “local signed difference in NdUnfilled”: Signed local difference in the number of unfilled d electrons in the ground state atom
- “local signed difference in NfUnfilled”: Signed local difference in the number of unfilled f electrons in the ground state atom
- “local signed difference in NUnfilled”: Signed local difference in the number of electrons in the ground state atom
- “local signed difference in GSbandgap”: Signed local difference in the ground state bandgap of the bulk material
- “maximum local difference in MendeleevNumber”: Maximum local difference in the atomic numbers
- “maximum local difference in Column”: Maximum local difference in the column (group) number
- “maximum local difference in Row”: Maximum local difference in the row number (period)
- “maximum local difference in Electronegativity”: Maximum local difference in electronegativity
- “maximum local difference in NsValence”: Maximum local difference in the number of s electrons in the ground state atom
- “maximum local difference in NpValence”: Maximum local difference in the number of p electrons in the ground state atom
- “maximum local difference in NdValence”: Maximum local difference in the number of d electrons in the ground state atom

- “maximum local difference in NfValence”: Maximum local difference in the number of f electrons in the ground state atom
- “maximum local difference in NValence”: Maximum local difference in the number of valence electrons in the ground state atom
- “maximum local difference in NsUnfilled”: Maximum local difference in the number of unfilled s electrons in the ground state atom
- “maximum local difference in NpUnfilled”: Maximum local difference in the number of unfilled p electrons in the ground state atom
- “maximum local difference in NdUnfilled”: Maximum local difference in the number of unfilled d electrons in the ground state atom
- “maximum local difference in NfUnfilled”: Maximum local difference in the number of unfilled f electrons in the ground state atom
- “maximum local difference in NUnfilled”: Maximum local difference in the number of unfilled electrons in the ground state atom
- “maximum local difference in GSbandgap”: Maximum local difference in the ground state bandgap of the bulk material
- “mimum local difference in MendeleevNumber”: Minimum local difference in the atomic numbers
- “mimum local difference in Column”: Minimum local difference in the column (group) number
- “mimum local difference in Row”: Minimum local difference in the row number (period)
- “mimum local difference in Electronegativity”: Minimum local difference in electronegativity
- “mimum local difference in NsValence”: Minimum local difference in the number of s electrons in the ground state atom
- “mimum local difference in NpValence”: Minimum local difference in the number of p electrons in the ground state atom
- “mimum local difference in NdValence”: Minimum local difference in the number of d electrons in the ground state atom
- “mimum local difference in NfValence”: Minimum local difference in the number of f electrons in the ground state atom
- “mimum local difference in NValence”: Minimum local difference in the number of valence electrons in the ground state atom
- “mimum local difference in NsUnfilled”: Minimum local difference in the number of unfilled s electrons in the ground state atom
- “mimum local difference in NpUnfilled”: Minimum local difference in the number of unfilled p electrons in the ground state atom
- “mimum local difference in NdUnfilled”: Minimum local difference in the number of unfilled d electrons in the ground state atom
- “mimum local difference in NfUnfilled”: Minimum local difference in the number of unfilled f electrons in the ground state atom
- “mimum local difference in NUnfilled”: Minimum local difference in the number of unfilled electrons in the ground state atom
- “mimum local difference in GSbandgap”: Minimum local difference in the ground state bandgap of the bulk material

Metal features

- “column”: Group number of the metal
- “row”: Period number of the metal
- “valenceelectrons”: Number of valence electrons of the ground state of the metal
- “diff18electrons”: Heuristic for missing electrons to a stable d-shell for the ground state of the metal
- “sunfilled”: Number of unfilled s electrons of the ground state metal
- “punfilled”: Number of unfilled p electrons of the ground state metal
- “dunfilled”: Number of unfilled d electrons of the ground state metal

Geometry features The “wt” order parameter describes how consistent a site is with a certain coordination number.

- “wt CN_1”: consistency of site with coordination number 1
- “sgl_bd CN_1”: order parameter for single bounded
- “wt CN_2”: consistency of site with coordination number 2
- “L-shaped CN_2”: order parameter for L-shaped coordination
- “water-like CN_2”: order parameter for water-shaped coordination
- “bent 120 degrees CN_2”: order parameter for 120° coordination
- “bent 150 degrees CN_2”: order parameter for 150° coordination
- “linear CN_2”: order parameter for linear coordination
- “wt CN_3”: consistency of site with coordination number 3
- “trigonal planar CN_3”: order parameter for trigonal planar coordination
- “trigonal non-coplanar CN_3”: order parameter for trigonal non-planar
- “T-shaped CN_3”: order parameter for T-shaped coordination
- “wt CN_4”: consistency of site with coordination number 4
- “square co-planar CN_4”: order parameter for square co-planar coordination
- “tetrahedral CN_4”: order parameter for tetrahedral coordination
- “rectangular see-saw-like CN_4”: order parameter for rectangular coordination
- “see-saw-like CN_4”: order parameter for see-saw coordination
- “trigonal pyramidal CN_4”: order parameter for trigonal pyramidal
- “wt CN_5”: consistency of site with coordination number 5
- “pentagonal planar CN_5”: order parameter for pentagonal planar coordination
- “square pyramidal CN_5”: order parameter for square pyramidal coordination
- “trigonal bipyramidal CN_5”: order parameter for trigonal bipyramidal coordination
- “wt CN_6”: consistency of site with coordination number 6
- “hexagonal planar CN_6”: order parameter for hexagonal planar coordination

- “octahedral CN_6”: order parameter for octahedral coordination
- “pentagonal pyramidal CN_6”: order parameter for pentagonal pyramidal coordination
- “wt CN_7”: consistency of site with coordination number 7
- “hexagonal pyramidal CN_7”: order parameter for hexagonal pyramidal coordination
- “pentagonal bipyramidal CN_7”: order parameter for pentagonal bipyramidal coordination
- “wt CN_8”: consistency of site with coordination number 8
- “body-centered cubic CN_8”: order parameter for body-centered cubic coordination
- “hexagonal bipyramidal CN_8”: order parameter for hexagonal bipyramidal coordination
- “wt CN_9”: consistency of site with coordination number 9
- “wt CN_10”: consistency of site with coordination number 10
- “wt CN_11”: consistency of site with coordination number 11
- “wt CN_12”: consistency of site with coordination number 12
- “cuboctahedral CN_12”: order parameter for cuboctahedral coordination
- “wt CN_13”: consistency of site with coordination number 13
- “wt CN_14”: consistency of site with coordination number 14
- “wt CN_15”: consistency of site with coordination number 15
- “wt CN_16”: consistency of site with coordination number 16
- “wt CN_17”: consistency of site with coordination number 17
- “wt CN_18”: consistency of site with coordination number 18
- “wt CN_19”: consistency of site with coordination number 19
- “wt CN_20”: consistency of site with coordination number 20
- “wt CN_21”: consistency of site with coordination number 21
- “wt CN_22”: consistency of site with coordination number 22
- “wt CN_23”: consistency of site with coordination number 23
- “wt CN_24”: consistency of site with coordination number 24
- “q4 CN_9”: Steinhardt bond orientational order parameter of order 4
- “q6 CN_9”: Steinhardt bond orientational order parameter of order 6
- “q2 CN_10”: Steinhardt bond orientational order parameter of order 2
- “q4 CN_10”: Steinhardt bond orientational order parameter of order 4
- “q6 CN_10”: Steinhardt bond orientational order parameter of order 6
- “q2 CN_11”: Steinhardt bond orientational order parameter of order 2
- “q4 CN_11”: Steinhardt bond orientational order parameter of order 4
- “q6 CN_11”: Steinhardt bond orientational order parameter of order 6
- “q2 CN_12”: Steinhardt bond orientational order parameter of order 2
- “q4 CN_12”: Steinhardt bond orientational order parameter of order 4
- “q6 CN_12”: Steinhardt bond orientational order parameter of order 12

D.3.2 Feature importance quantification

Model performance with different feature sets

One direct way to measure the benefit of adding additional features to a model is to compare the metrics for predictive performance, leaving the training (and test size) fixed and varying the feature set. We used the same number of training, validation, and holdout points in all cases, which we collected using submodular selection in the respective feature space.

Table 31: Performance for different feature sets. Performance metrics for models trained on different feature sets. The diverse set selection was performed separately on each feature set.

feature sets	accuracy	balanced accuracy	F ₁ micro	F ₁ macro	precision	recall
metal center, chemistry, geometry	0.99	0.98	0.99	0.99	0.99	0.99
RACs, metal centre	0.98	0.97	0.98	0.98	0.98	0.98
RACs, metal centre, chemistry, geometry	0.99	0.97	0.99	0.97	0.99	0.99
chemistry, geometry	0.99	0.97	0.99	0.97	0.99	0.99
RACs, chemistry, metal centre	0.98	0.97	0.98	0.96	0.98	0.98
RACs, chemistry	0.98	0.96	0.98	0.96	0.98	0.98
metal centre, chemistry	0.96	0.90	0.96	0.91	0.96	0.96
chemistry	0.96	0.89	0.96	0.90	0.96	0.96
geometry	0.91	0.84	0.91	0.86	0.91	0.91
RACs	0.88	0.80	0.88	0.84	0.88	0.88
metal centre	0.90	0.75	0.90	0.73	0.90	0.90

D.3.3 Model using an optimized feature set

Using insights from the feature importance analysis (see below), we constructed a model with a small and optimized feature set (`optimized_feature_set` preset in `mine_mof_ox`, see for example experiment `a0104584b5c0451aa631b8e516491e5f` on `comet.ml` for hyperparameter optimization, `1d2c686f9dc94007ae2f6aa9391af555` for the training). For this model, we used the local structural order parameters (without the Steinhardt bond orientational order parameters), local signed differences in electronegativity, and valence electrons in the shell of the group state atoms as well as the row and the column. The performance of this model is comparable to one with the full feature set, e.g., we still achieve an overall accuracy of over 98 %, and the AUC scores for all oxidation states are greater than 0.96. The confusion matrix is listed in Table 32, an overview of relevant metrics is listed in Tables 33 and 34.

D.3.4 Shapely additive explanations (SHAP) feature importance

To estimate the feature importance for the case studies, we used the SHapley Additive exPlanations (SHAP) method.²⁶⁹ Analyses using the permutation feature importance technique show the same qualitative results.

To eliminate the influence of the metal center features, we used only feature vectors for copper sites from the training set as the background data. For efficiency

Table 32: Confusion matrix for a holdout test set using a model with an optimized feature set.

Numbers in each cell show the number of classified metal centers for each case. In the ideal error-free case the matrix would only have entries on the diagonal. The confusion matrix shows all predictions, i.e., including high- and low-confidence predictions.

	prediction					
	I	II	III	IV	V	VI
ground truth	I	8172	50	0	0	0
	II	151	23953	81	0	0
	III	0	329	6734	14	0
	IV	0	0	48	1073	10
	V	0	0	0	4	284
	VI	0	0	0	0	1528

Table 33: Global performance metrics for the model with the optimized feature set.

metric	value
accuracy	0.9944 ± 0.003
F ₁ macro	0.97
F ₁ micro	0.98
κ	0.97 ± 0.002
κ _{no prevalence} ⁶⁰¹	0.97
relative classifier information ⁶⁰²	0.92
precision macro	0.98
precision micro	0.98

reasons, we summarize the background data using k -means clustering and weight the cluster centroids by the number of neighbors.

Feature importance plots in the text show the absolute values, $\|\phi_i\|$, of the SHAP values, which are defined as the sum over all feature subsets $S \subseteq F$, where F is the set of all features, using a model $f_{S \cup \{i\}}$ that is trained with feature i absent and one, f_S , with that feature present:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{\|S\|! (\|F\| - \|S\| - 1)!}{\|F\|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]. \quad (32)$$

In the case of global explanations, those values are averaged over multiple sites. In summary plots (e.g., Figure 126), the features are ordered according to overall importance (the most important features are plotted first).

The analysis of the feature importance (cf. images/oximachine 126 and 127) reveals that our model captures several chemical intuitions:

- Going to higher row numbers, the metals become more basic (more shielded valence electrons) and more able to attain higher oxidation states.⁶²⁰

Table 34: Summary of class statistics for the model with the optimized feature set. Decimal places for values > 0.99 are cut and not rounded.

metric	I	II	III	IV	V	VI
accuracy	0.99	0.99	0.99	0.99	0.99	0.99
adjusted F score	0.99	0.99	0.98	0.98	0.97	0.99
AUC	0.99	0.99	0.97	0.97	0.97	0.99
AUPR	0.99	0.99	0.97	0.97	0.95	0.99
precision	0.98	0.98	0.98	0.98	0.97	0.98
Matthews correlation coefficient ⁶⁰³	0.98	0.97	0.96	0.96	0.95	0.99
Gini index ⁶⁰⁴	0.99	0.97	0.95	0.94	0.93	0.99



Figure 126: SHAP summary plot of global model explanations. SHAP summary plot for prediction on the holdout test set (200 examples selected using submodular selection). The violins show the distributions of the SHAP values, and the color indicates the value of the feature. The SHAP value indicates the influence on the prediction of the model; a more positive value indicates that this particular feature value makes the model predict a higher oxidation state. Gray areas mark areas for which we do not have any data points.

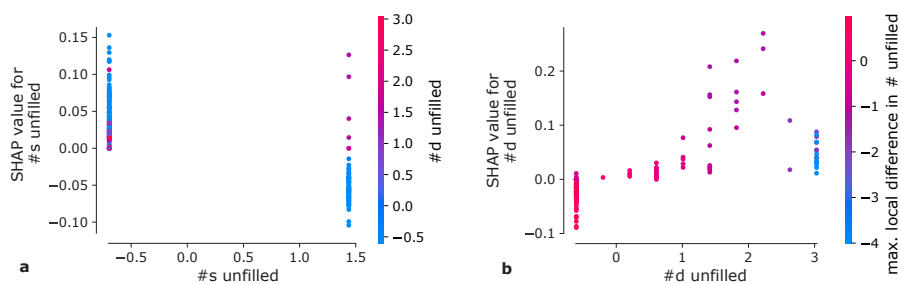


Figure 127: SHAP local interpretations. SHAP value as a function of the feature value, for the number of unfilled s (a) and d (b) electrons, respectively, and color-coded the feature with the strongest interaction.

- The highest oxidation states and the highest variation in oxidation states tend to occur in the middle of the d block (number of unfilled d electrons, column).⁶²¹
- Unfilled s electrons (alkali metals or some d block anomalies) influence the

predicted outcome differently depending on the number of d electrons.

- There are strong interactions between chemistry and geometry (the features that influence the SHAP value for geometry features most strongly are often chemistry features).

D.3.5 Cases of (in)correct classification by the bond-valence sum method

Comparison between SHAP and permutation feature importance

The variance analysis using PCA shows (cf. Section D.4) that the geometrical factors are most important for our model to distinguish between the two oxidation states of copper. As discussed above, we observe that our model generally associates higher coordination numbers with higher oxidation states—which follows chemical intuition. Especially the square pyramidal coordination order parameter seems to be a strong feature.



Figure 128: SHAP summary plot for the Cu MOFs in the holdout test set analyzed using the bond valence (BV) sum method. As background data, we used a *knn* summarized set of Cu sites from the training set. The violins show the distributions of the data, and the color indicates the value of the feature. The SHAP value indicates the influence on the prediction of the model; a more positive value indicates that this particular feature value makes the model predict a higher oxidation state. Gray areas mark areas for which we do not have any data points.

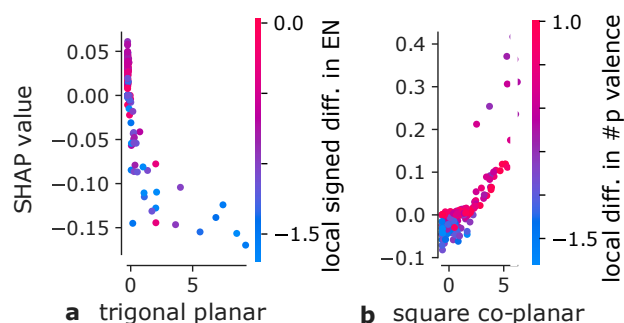


Figure 129: SHAP dependence plots for coordination number three (a) and four (b) order parameters, respectively.

D.3.6 Cu(I,II)-BTC

The SHAP summary plot is shown in Figure 130 and follows chemical intuition: High similarity to highly coordinated coordination environments (square pyramidal, trigonal pyramidal) increases the model output, i.e., the predicted oxidation states whereas high similarity to coordination environments with fewer neighbors (linear, 15° bent) decrease the predicted oxidation state. Similarly, we find that a higher local difference in electronegativity (or local difference in p valence electrons) makes our model predict a higher oxidation state.

The absolute SHAP values for all features of both the Cu sites are listed in Table 35.

Table 35: SHAP values for Cu(I/II)-BTC.. Listing of SHAP values for all features, for the macrocyclic and paddlewheel.

feature	Cu(I)	Cu(II)
square pyramidal CN 5	-0.909072	0.613580
linear CN 2	-0.749447	0.004850
bent 150 degrees CN 2	-0.441103	-0.001509
wt CN 5	-0.417583	0.267922
wt CN 2	-0.317646	0.015357
local signed difference in # p valence	-0.259301	0.085554
square co-planar CN 4	-0.219004	0.004310
octahedral CN 6	-0.210336	0.012484
trigonal bipyramidal CN 5	-0.172772	0.143708
local signed difference in electronegativity	-0.164075	0.097353
wt CN 6	-0.154959	0.014733
local difference in # d valence	-0.136092	0.031119
local difference in # s valence	-0.108644	0.028151
pentagonal planar CN 5	-0.105035	0.015531
rectangular see-saw-like CN 4	-0.101144	0.021873
minimum local difference in Mendeleev number	-0.093163	-0.002152
local signed difference in # s valence	-0.089992	0.055203
maximum local difference in # d valence	-0.070762	0.013815
minimum local difference in column	-0.070761	0.003338
hexagonal planar CN 6	-0.069904	0.012230
local difference in electronegativity	-0.058395	0.156434

Continued on next page

Table 35: SHAP values for Cu(I/II)-BTC.. Listing of SHAP values for all features, for the macrocyclus and paddlewheel.

feature	Cu(I)	Cu(II)
local difference in # p unfilled	-0.047810	0.012924
maximum local difference in row	-0.046069	0.003691
local signed difference in Mendelev number	-0.041669	0.000337
wt CN 8	-0.038648	0.016203
trigonal non-coplanar CN 3	-0.035160	0.075156
pentagonal bipyramidal CN 7	-0.035019	0.000838
hexagonal bipyramidal CN 8	-0.034717	0.000000
local signed difference in column	-0.031195	0.012950
trigonal pyramidal CN 4	-0.030897	0.019953
wt CN 9	-0.030870	0.011095
local difference in # p valence	-0.030326	0.031584
pentagonal pyramidal CN 6	-0.023506	0.015684
maximum local difference in Mendelev number	-0.023378	0.003170
local signed difference in # unfilled	-0.023308	0.009563
hexagonal pyramidal CN 7	-0.021975	0.013250
local difference in column	-0.021620	0.004096
maximum local difference in electronegativity	-0.019064	0.057823
minimum local difference in Gs bandgap	-0.019059	0.004106
maximum local difference in # unfilled	-0.018811	0.011059
bent 120 degrees CN 2	-0.017309	0.016965
wt CN 14	-0.015547	0.010130
local difference in nunfilled	-0.015272	0.001980
body-centered cubic CN 8	-0.015137	0.000000
local difference in # f valence	-0.014620	0.017249
local signed difference in # p unfilled	-0.014207	0.003621
maximum local difference in # valence	-0.013693	0.010710
maximum local difference in Gs bandgap	-0.013342	0.006750
single bond CN 1	-0.012870	0.014012
maximum local difference in # p valence	-0.011895	0.038656
wt CN 4	-0.011703	0.006224
minimum local difference in # valence	-0.011239	0.006032
local difference in # d unfilled	-0.010490	0.014419
water-like CN 2	-0.006790	0.033701
cuboctahedral CN 12	-0.006654	0.011608
local signed difference in # valence	-0.006469	0.017717
l-shaped CN 2	-0.005743	0.022772
local signed difference in # d valence	-0.001617	0.006144
local signed difference in # d unfilled	-0.000048	0.003249
minimum local difference in # f unfilled	0.000000	0.000000
wt CN 15	0.000000	0.000000
wt CN 16	0.000000	0.000000
wt CN 17	0.000000	0.000000
wt CN 18	0.000000	0.000000
wt CN 19	0.000000	0.000000
wt CN 20	0.000000	0.000000
wt CN 21	0.000000	0.000000
wt CN 22	0.000000	0.000000
wt CN 13	0.000000	0.000000
wt CN 11	0.000000	0.000000
wt CN 24	0.000000	0.000000
d unfilled	0.000000	0.000000
maximum local difference in # d unfilled	0.000000	0.000000
maximum local difference in # f unfilled	0.000000	0.000000
maximum local difference in # s valence	0.000000	0.000000
local signed difference in # f unfilled	0.000000	0.000000
minimum local difference in electronegativity	0.000000	0.019048
minimum local difference in # s valence	0.000000	0.000000
minimum local difference in # p valence	0.000000	0.000000
minimum local difference in # d valence	0.000000	0.000000
minimum local difference in # f valence	0.000000	0.000000

Continued on next page

Table 35: SHAP values for Cu(I/II)-BTC.. Listing of SHAP values for all features, for the macrocyclus and paddlewheel.

feature	Cu(I)	Cu(II)
minimum local difference in # s unfilled	0.000000	0.000000
minimum local difference in # p unfilled	0.000000	0.000000
minimum local difference in # d unfilled	0.000000	0.000000
wt CN 23	0.000000	0.000000
local signed difference in row	0.000000	0.007132
minimum local difference in # unfilled	0.000000	0.000000
local difference in Gs bandgap	0.000000	0.034698
column	0.000000	0.000000
row	0.000000	0.000000
local difference in # funfilled	0.000000	0.000000
valence electrons	0.000000	0.000000
local difference in # valence	0.000000	0.012122
diff to 18 electrons	0.000000	0.000000
s unfilled	0.000000	0.000000
p unfilled	0.000000	0.000000
maximum local difference in # f valence	0.000000	0.000000
maximum local difference in # s unfilled	0.000000	0.000000
local difference in row	0.000270	0.002022
local difference in Mendeleev number	0.003694	0.011932
wt CN 10	0.004626	0.004663
t-shaped CN 3	0.005452	0.019825
wt CN 1	0.007540	0.004919
tetrahedral CN 4	0.011486	0.047986
maximum local difference in # p unfilled	0.011745	0.001457
local signed difference in # f valence	0.011933	0.000000
wt CN 3	0.012414	0.007049
minimum local difference in row	0.013717	0.000000
wt CN 12	0.014530	0.003594
wt CN 7	0.016616	0.007560
local difference in # s unfilled	0.017490	0.009048
maximum local difference in column	0.020533	0.020929
trigonal planar CN 3	0.021260	0.237176
see-saw-like CN 4	0.039270	0.025903
local signed difference in Gs bandgap	0.041773	0.033488
local signed difference in # s unfilled	0.075302	-0.017143

D.3.7 MIL-47

The SHAP values for all features of both the MIL-47(as) and the activated structure are listed in Table 36.

Table 36: SHAP values for MIL-47. Listing of SHAP values for all features, for the activated and the as-synthesised (as) structure.

feature	MIL-47 (as)	MIL-47 (activated)
octahedral CN 6	1.549734	0.254080
wt CN 6	0.686747	0.051034
pentagonal pyramidal CN 6	0.618746	0.051499
tetrahedral CN 4	0.603557	0.323607
wt CN 1	0.401267	0.007937
minimum local difference in column	0.331678	0.117328
single bond CN 1	0.260923	0.167693
pentagonal planar CN 5	0.242957	0.091866
minimum local difference in # valence	0.208751	0.011380
water-like CN 2	0.200464	0.069949
minimum local difference in electronegativity	0.194526	0.074362
hexagonal planar CN 6	0.176997	0.019653
square pyramidal CN 5	0.168113	0.079079

Continued on next page

Table 36: SHAP values for MIL-47. Listing of SHAP values for all features, for the activated and the as-synthesised (as) structure.

feature	MIL-47 (as)	MIL-47 (activated)
maximum local difference in Gs bandgap	0.156312	0.009012
trigonal pyramidal CN 4	0.155963	0.104752
trigonal bipyramidal CN 5	0.144059	0.003517
wt CN 5	0.097325	0.009215
wt CN 4	0.093918	0.088117
trigonal non-coplanar CN 3	0.082951	0.056379
bent 120 degrees CN 2	0.055490	0.014857
local signed difference in # valence	0.054438	0.139427
maximum local difference in # unfilled	0.048549	0.018605
square co-planar CN 4	0.045803	0.024420
see-saw-like CN 4	0.041538	0.037883
L-shaped CN 2	0.036103	0.014675
local signed difference in Gs bandgap	0.035061	0.007301
maximum local difference in column	0.034825	0.013176
local difference in row	0.033432	0.053456
maximum local difference in # d valence	0.028875	0.005876
minimum local difference in # p valence	0.021635	0.050926
maximum local difference in electronegativity	0.021287	0.021878
maximum local difference in # valence	0.017577	0.029955
minimum local difference in Mendelev number	0.017080	0.000000
local signed difference in electronegativity	0.016949	0.000000
local difference in Mendelev number	0.016421	0.011915
local difference in Gs bandgap	0.015905	0.012549
body-centered cubic CN 8	0.015069	0.008068
local difference in # d valence	0.014898	0.000000
wt CN 2	0.012992	0.001018
local difference in # unfilled	0.012812	0.011292
wt CN 3	0.011915	0.018215
linear CN 2	0.010913	0.005758
hexagonal pyramidal CN 7	0.010699	0.025266
local signed difference in Mendelev number	0.009299	0.023733
local difference in # p unfilled	0.007539	0.009860
local difference in column	0.005591	0.012662
local difference in electronegativity	0.004472	0.035064
wt CN 7	0.002071	0.012288
trigonal planar CN 3	0.000428	0.005401
wt CN 16	0.000000	0.000000
T-shaped CN 3	0.000000	0.006251
wt CN 8	0.000000	0.027586
hexagonal bipyramidal CN 8	0.000000	0.009893
wt CN 23	0.000000	0.000000
wt CN 22	0.000000	0.000000
bent 150 degrees CN 2	0.000000	0.011127
wt CN 9	0.000000	0.010256
wt CN 10	0.000000	0.000000
wt CN 11	0.000000	0.000000
wt CN 21	0.000000	0.000000
wt CN 15	0.000000	0.000000
wt CN 12	0.000000	0.000000
wt CN 20	0.000000	0.000000
wt CN 19	0.000000	0.000000
rectangular see-saw-like CN 4	0.000000	0.000000
cuboctahedral CN 12	0.000000	0.000000
pentagonal bipyramidal CN 7	0.000000	0.015977
wt CN 17	0.000000	0.000000
wt CN 13	0.000000	0.000000
wt CN 14	0.000000	0.000000
wt CN 18	0.000000	0.000000
minimum local difference in # unfilled	0.000000	0.026772
d unfilled	0.000000	0.000000

Continued on next page

Table 36: SHAP values for MIL-47. Listing of SHAP values for all features, for the activated and the as-synthesised (as) structure.

feature	MIL-47 (as)	MIL-47 (activated)
p unfilled	0.000000	0.000000
maximum local difference in Mendeleev number	0.000000	0.034787
local signed difference in # unfilled	0.000000	0.008021
local signed difference in # f unfilled	0.000000	0.000000
local signed difference in # d unfilled	0.000000	0.021360
local signed difference in # p unfilled	0.000000	0.019717
local signed difference in # s unfilled	0.000000	0.013731
local signed difference in # f valence	0.000000	0.000000
local signed difference in # d valence	0.000000	0.007560
local signed difference in # p valence	0.000000	0.029144
local signed difference in # s valence	0.000000	0.007232
local signed difference in row	0.000000	0.042591
local signed difference in column	0.000000	0.028065
local difference in # f unfilled	0.000000	0.000000
local difference in # d unfilled	0.000000	0.021614
local difference in # s unfilled	0.000000	0.007000
local difference in # valence	0.000000	0.015774
local difference in # f valence	0.000000	0.000000
local difference in # p valence	0.000000	0.004856
local difference in # s valence	0.000000	0.033869
maximum local difference in row	0.000000	0.005906
maximum local difference in # s valence	0.000000	0.000000
maximum local difference in # p valence	0.000000	0.023295
minimum local difference in # p unfilled	0.000000	0.012704
s unfilled	0.000000	0.000000
diff to 18 electrons	0.000000	0.000000
valence electrons	0.000000	0.000000
row	0.000000	0.000000
column	0.000000	0.000000
minimum local difference in Gs bandgap	0.000000	0.000000
minimum local difference in # f unfilled	0.000000	0.000000
minimum local difference in # d unfilled	0.000000	0.000000
minimum local difference in # s unfilled	0.000000	0.000000
maximum local difference in # f valence	0.000000	0.000000
minimum local difference in # f valence	0.000000	0.000000
minimum local difference in # d valence	0.000000	0.000000
minimum local difference in # s valence	0.000000	0.016594
minimum local difference in row	0.000000	0.023644
maximum local difference in # f unfilled	0.000000	0.000000
maximum local difference in # d unfilled	0.000000	0.024568
maximum local difference in # p unfilled	0.000000	0.020117
maximum local difference in # s unfilled	0.000000	0.090870
wt CN 24	0.000000	0.000000

D.4 PRINCIPAL COMPONENT ANALYSIS FOR THE COPPER CASE STUDY

To extract the common features of the cases in which the BV sum fails (cf. section D.7), we performed Huber regression ($\epsilon = 3$, green line in Figure 131) on the two-dimensional principal component (PC) embedding of the feature space (which can explain 37.4 % of the total variance of the data, note that this analysis is linear) of the copper sites for which the BV sum method predicted the wrong oxidation state. We then extracted the features with the highest loading by taking the dot product with the principal components. In this way, we identified the five features with the highest loading to be features for coordination order parameters of coordination number four. Note that in this analysis, the metal center features automatically vanish as we



Figure 130: Extended SHAP summary plot for the case of Cu(I,II)-BTC. The violins show the distributions of the data, and the color indicates the value of the feature. The SHAP value indicates the influence on the prediction of the model; a more positive value indicates that this particular feature value makes the model predict a higher oxidation state. Gray areas mark areas for which we do not have any data points.

limit our attention to only copper centers.

The fact that the order parameters for coordination geometry are powerful in separating the oxidation states is also evident from Figure 132, where we color the points depending on the oxidation state. It is clearly observable that a clear separation between the two oxidation states is possible mostly by means of the geometrical order parameters.

D.5 DETAILS ABOUT THE TEST SETS

D.5.1 Detailed analysis of the predictive performance

Table 37: Classification metrics on the test set as a function of the metal. Micro and macro refer to the averaging methods, where micro averaging gives high weights to rare classes.

metal	accuracy	recall (micro)	precision (micro)	recall (macro)	precision (macro)
Ag	0.999	0.999	0.999	0.5	0.499

Continued on next page

Table 37: Classification metrics on the test set as a function of the metal. Micro and macro refer to the averaging methods, where micro averaging gives high weights to rare classes.

metal	accuracy	recall (micro)	precision (micro)	recall (macro)	precision (macro)
Am	1.0	1.0	1.0	1.0	1.0
Au	0.989	0.989	0.989	0.75	0.995
Ba	1.0	1.0	1.0	1.0	1.0
Bi	1.0	1.0	1.0	1.0	1.0
Ca	1.0	1.0	1.0	1.0	1.0
Cd	1.0	1.0	1.0	1.0	1.0
Ce	0.931	0.931	0.931	0.5	0.466
Co	0.959	0.959	0.959	0.646	0.747
Cr	0.954	0.954	0.954	0.604	0.667
Cs	1.0	1.0	1.0	1.0	1.0
Cu	0.979	0.979	0.979	0.981	0.972
Dy	1.0	1.0	1.0	1.0	1.0
Er	0.995	0.995	0.995	0.5	0.498
Eu	0.978	0.978	0.978	0.579	0.989
Fe	0.894	0.894	0.894	0.886	0.878
Ga	1.0	1.0	1.0	1.0	1.0
Gd	1.0	1.0	1.0	1.0	1.0
Hf	1.0	1.0	1.0	1.0	1.0
Hg	0.979	0.979	0.979	0.5	0.49
Ho	1.0	1.0	1.0	1.0	1.0
In	1.0	1.0	1.0	1.0	1.0
Ir	1.0	1.0	1.0	1.0	1.0
K	1.0	1.0	1.0	1.0	1.0
La	1.0	1.0	1.0	1.0	1.0
Li	1.0	1.0	1.0	1.0	1.0
Lu	1.0	1.0	1.0	1.0	1.0
Mg	1.0	1.0	1.0	1.0	1.0
Mn	0.961	0.961	0.961	0.884	0.921
Mo	0.985	0.985	0.985	0.796	0.785
Na	1.0	1.0	1.0	1.0	1.0
Nb	0.828	0.828	0.828	0.25	0.25
Nd	1.0	1.0	1.0	1.0	1.0
Ni	1.0	1.0	1.0	1.0	1.0
Np	1.0	1.0	1.0	1.0	1.0
Os	0.889	0.889	0.889	0.444	0.5
Pb	1.0	1.0	1.0	1.0	1.0
Pd	1.0	1.0	1.0	1.0	1.0
Pr	1.0	1.0	1.0	1.0	1.0
Pt	0.916	0.916	0.916	0.5	0.458
Pu	0.0	0.0	0.0	0.0	0.0
Rb	1.0	1.0	1.0	1.0	1.0
Re	0.0	0.0	0.0	0.0	0.0
Rh	0.913	0.913	0.913	0.905	0.957
Ru	0.938	0.938	0.938	0.959	0.894
Sc	1.0	1.0	1.0	1.0	1.0
Sm	1.0	1.0	1.0	1.0	1.0
Sn	0.958	0.958	0.958	0.604	0.657
Sr	1.0	1.0	1.0	1.0	1.0
Tb	1.0	1.0	1.0	1.0	1.0
Th	1.0	1.0	1.0	1.0	1.0
Ti	1.0	1.0	1.0	1.0	1.0
Tl	0.978	0.978	0.978	0.917	0.987
Tm	1.0	1.0	1.0	1.0	1.0
U	1.0	1.0	1.0	1.0	1.0
V	0.942	0.942	0.942	0.893	0.967
W	0.992	0.992	0.992	0.778	0.938
Y	1.0	1.0	1.0	1.0	1.0
Yb	0.95	0.95	0.95	0.5	0.475
Zn	1.0	1.0	1.0	1.0	1.0

Continued on next page

Table 37: Classification metrics on the test set as a function of the metal. Micro and macro refer to the averaging methods, where micro averaging gives high weights to rare classes.

metal	accuracy	recall (micro)	precision (micro)	recall (macro)	precision (macro)
Zr	1.0	1.0	1.0	1.0	1.0

D.5.2 Examples with spectroscopic evidence

In addition to a larger holdout set which we used to estimate test statistics, we also used some examples for which we found spectroscopic evidence in the literature as a separate test set. To find those cases, we performed literature research using the terms MOF + {XPS, NEXFAS, EXFAS, XANES, redox, oxidation state, susceptibility, magnetometer}.

We considered only those cases in which we could retrieve a CIF and list them in Table 38. In cases where the structure is deposited in the CSD, we excluded the structures from the training set.

Table 38: Test structures with strong experimental evidence for the oxidation state assignment.

CSD reference code	metal (ox. st.)	predictions	assignment technique
KOLVOC ⁶²²	Fe(II)	Fe(II)	Moessbauer
XUVDEZ ⁶²³	Fe(II)	Fe(II)	Moessbauer
WODZEX ⁶²⁴	Fe(II)	Fe(II)	Moessbauer
MAHSUK01 ⁶²⁵	Co(II)	Co(II)	X-ray photoelectron spectroscopy (XPS)
BUPVEP ⁶²⁶	Ce(IV)	Ce(III)	X-ray absorption near edge structure (XANES)
ZIFTIU ⁶²⁷	Ce(III)	Ce(III)	XANES
ZITMUN ⁶²⁸	Ce(III)	Ce(III)	XANES
KAJZIH ³³³	Cu(I/II)	Cu(I/II)	XPS
ORIVUI ⁶²⁹	Ni(II)	Ni(II)	XPS
QAMTEG ⁶³⁰	Tb(III)	Tb(III)	XPS
COKNOH01 ³⁴⁰	Fe(II)	Fe(II)	Moessbauer
GUVZEE ³⁴⁰	Fe(III)	Fe(III)	Moessbauer
GUVZII ³⁴⁰	Fe(II)	Fe(II)	Moessbauer
GASMUK ⁶³¹	Au(III)	Au(III)	nuclear magnetic resonance (NMR)
DOVBIB ⁶³²	Co(III)	Co(III)	magnetic susceptibility
JIZJAF ⁶³³	Co(II)	Co(II)	magnetometer
JIZJE ⁶³³	Co(II)	Co(II)	magnetometer
JIZJIN ⁶³³	Co(II)	Co(II)	magnetometer
JIZJOT ⁶³³	Co(II)	Co(II)	magnetometer
JIZJUZ ⁶³³	Co(II)	Co(II)	magnetometer
PETWOC ⁶³⁴	U(IV)	U(IV)	XPS
PETWUI ⁶³⁴	U(IV)	U(IV)	XPS
PETXAP ⁶³⁴	U(IV)	U(IV)	XPS
VIMRAM ⁶³⁵	Cu(II)	Cu(II)	magnetometer
YOPQEB ⁶³⁶	Co(II)	Co(II)	magnetometer
RISXAU ⁶³⁷	Fe(II)	Fe(II)	magnetometer
DOTVEP ⁶³⁸	Co(II)	Co(II)	magnetometer
DOTVOZ ⁶³⁸	Co(II)	Mn(II)	magnetometer
DOTVUF ⁶³⁸	Mn(II)	Mn(II)	magnetometer
DOTVAM ⁶³⁸	Co(II)	Co(II)	magnetometer
VAYGAF ⁶³⁹	Co(II)	Co(II)	magnetometer
QUBBEU ⁶⁴⁰	Co(II)	Co(II)	magnetometer
KIDLAL ⁶⁴¹	Mn(II)	Mn(II)	magnetometer
ACODEE ⁶⁴²	Fe(II)	Fe(II)	Moessbauer
YOCTIW ⁶⁴³	Cu(I/II)	Cu(I/II)	XPS

Continued on next page

Table 38: Test structures with strong experimental evidence for the oxidation state assignment.

CSD reference code	metal (ox. st.)	predictions	assignment technique
BAHLED ⁶⁴⁴	Fe(II)	Fe(II)	magnetometer
TINXAR ⁶⁴⁵	Co(II)	Co(II)	magnetometer
XOGORO ⁶⁴⁶	Mn(II)	Mn(II)	magnetometer
GIFMIU ⁶⁴⁷	Fe(II, III)	Fe(II, III)	magnetometer
CIXNUV ⁶⁴⁸	Mn(II)	Mn(II)	magnetometer
CIXPAD ⁶⁴⁸	Mn(II)	Mn(II)	magnetometer
YIGXOD ⁶⁴⁹	Co(II)	Co(II)	magnetometer
ACUWIH ⁶⁵⁰	Mn(II)	Mn(II)	XPS, magnetometer
WCUWON ⁶⁵⁰	Co(II)	Co(II)	XPS, magnetometer
MOHSEJ ⁶⁵¹	Co(II, III)	Co(II, III)	magnetometer
DEFRIH ⁶⁵²	Gd(III)	Gd(III)	XPS
TISHAF ⁶⁵³	Fe(III)	Fe(III)	magnetometer
GUJREK ⁶⁵⁴	Cu(I/II)	Cu(I/II)	magnetometer, electron paramagnetic resonance (EPR)
YADHIV ⁶⁵⁵	Cu(II)	Cu(II)	magnetometer
YADHOB ⁶⁵⁵	Cu(II)	Cu(II)	magnetometer
YADHUH ⁶⁵⁵	Cu(II)	Cu(II)	magnetometer

D.5.3 Test on structures added in a new CSD release

The training and testing of the models was carried out using the CSD release from May 2019. With the release from November 2019, new structures were added to the MOF subset which our model could never see before. For this reason, the new additions are a good test for our model.

There were 1166 additions (structures with new reference code) in the non-disordered set, for 952 of which the chemical name in the CSD contains an oxidation state. In total, we analyzed 5587 metal sites for these new additions.

We find disagreement between the oxidation state in the chemical name in the CSD and our prediction for 92 metal sites, corresponding to 13 unique structures. Overall, this corresponds to an accuracy of >99%. Also the area under the receiver-operating characteristic curve (AUC-ROC) is greater 0.94 for all classes except V, i.e., I, II, III, IV, VI. The F_1 macro score is 0.81, the F_1 micro score is 0.98. The confusion matrix is shown in Table 39

Also, in this small analysis, we found typos in the CSD, which also list in Table 39, all four are copper compounds (HOJHUM, HOMQEI, NOJSUD, SOKKAH).

Table 39: Confusion matrix for test on the new additions to the CSD. Numbers in each cell show the number of classified metal centers for each case. In the ideal error-free case, the matrix would only have entries on the diagonal.

		prediction					
ground truth		I	II	III	IV	V	VI
	I	354	46	0	0	0	0
	II	12	3678	14	0	0	0
	III	0	20	713	0	0	0
	IV	0	0	0	637	0	0
	V	0	0	0	2	0	0
	VI	0	0	0	0	0	111

D.5.4 Transfer to COFs

Additionally, we also tested the transferability of our model to COFs for which we found some examples with metals in the CURATED COFs database.¹⁴⁷ We only considered COFs for our test case for which the oxidation state of the metal center was

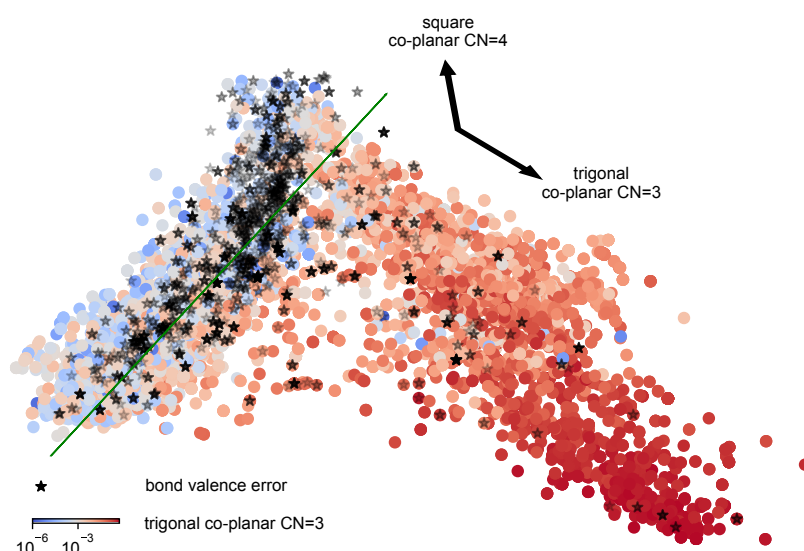


Figure 131: Projection of the feature space onto the two first principal components (linear combinations of features that capture most of the variance in the data). The color coding shows the value of the trigonal co-planar feature.

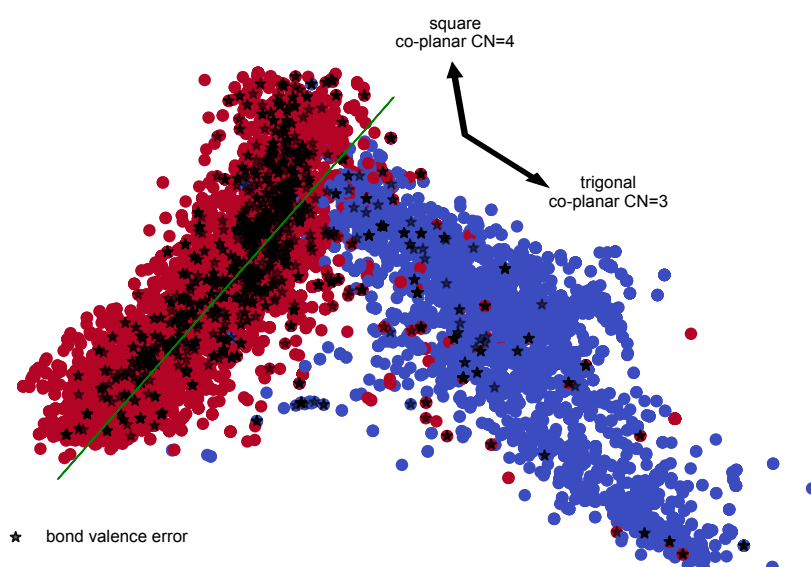


Figure 132: Projection of the feature space onto the two first principal components (linear combinations of features that capture most of the variance in the data). The color encoding shows the oxidation states. The green line is obtained from Huber regression on the erroneous cases, which are denoted by stars.

explicitly mentioned in either the supporting information or the paper's main text. The compounds and oxidation states of the metal centers are listed in Table 40. Our model correctly predicted all cases with high confidence (all except vanadium).

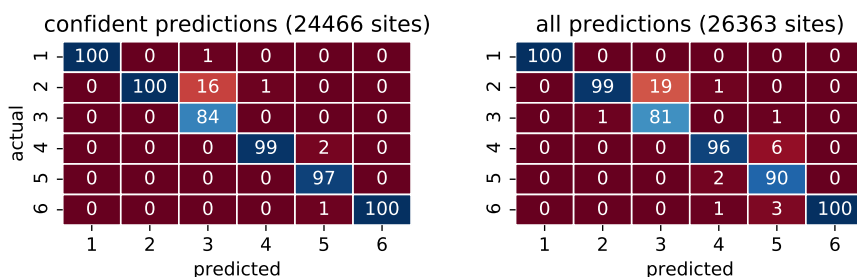


Figure 133: Confusion matrix for only elements of the d-block.

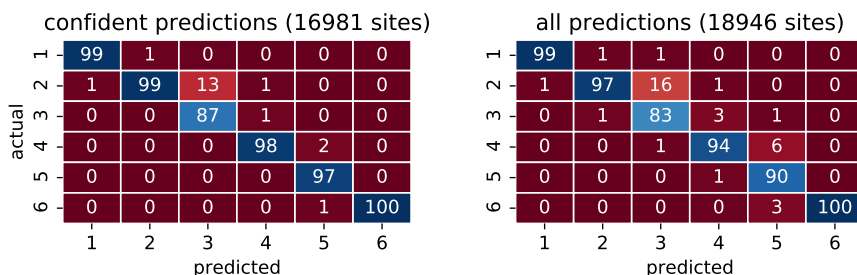


Figure 134: Confusion matrix for only elements for which there are at least two oxidation states for which the least frequent one occurs with at least 5 % frequency in the MOF subset of the CSD.

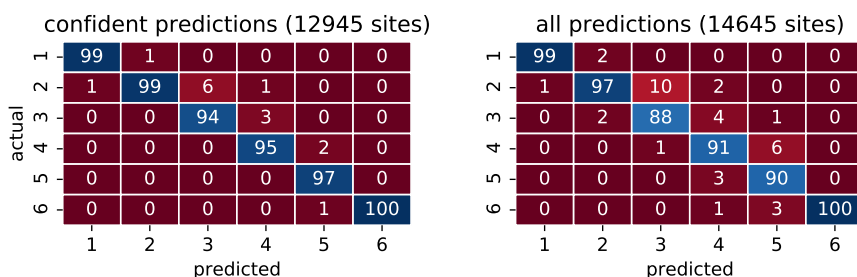


Figure 135: Confusion matrix for only elements for which there are at least two oxidation states for which the least frequent one occurs with at least 10 % frequency in the MOF subset of the CSD.

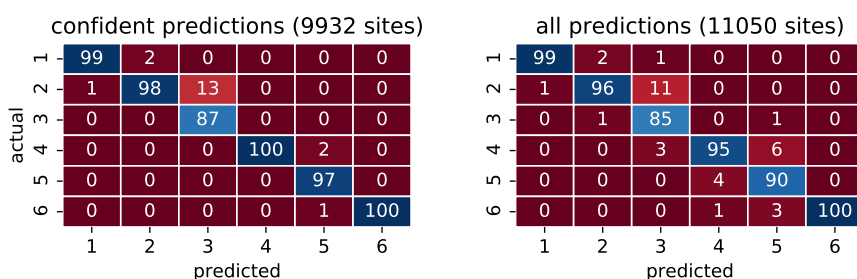


Figure 136: Confusion matrix for only elements for which there are at least two oxidation states for which the least frequent one occurs with at least 20 % frequency in the MOF subset of the CSD.

Table 40: Transfer to COFs. COFs with metals in the ligand for which the oxidation state is reported in the paper.

CURATED COF id	metal (oxidation state)
11010N2 ⁶⁵⁶	Ni(II)
12061N2 ⁶⁵⁷	Cu(II)
12062N2 ⁶⁵⁷	Co(II)
13110N2 ⁶⁵⁸	Cu(II)
15180N2 ⁶⁵⁹	Cu(II)
15181N2 ⁶⁵⁹	Cu(II)
15182N2 ⁶⁵⁹	Cu(II)
18080N2 ⁶⁶⁰	Co(II)
18081N2 ⁶⁶⁰	Co(II)
18082N2 ⁶⁶⁰	Co(II)
18083N2 ⁶⁶⁰	Co(II)
19040N2 ⁶⁶¹	V(IV)
19041N2 ⁶⁶¹	V(IV)
19270N2 ⁶⁶²	Zn(II)
19271N2 ⁶⁶²	Zn(II)

D.5.5 Transfer to transition metal complexes

As an additional test for transferability, we considered a collection of small transition metal complexes of Mn, Fe, Co, and Cr in oxidation states II and III that Janet and Kulik extracted from the CSD and optimized using hybrid-level DFT and used to test their NN as it “spans a broader range of compounds”.³⁴¹ We summarize the oxidation states of the metal centers in those structures along with the predictions of our model in Table 41. Note that we used the geometry of the spin state with the lowest energy for this test.

The global performance statistics are summarized in Table 42 and the class statistics in Table 43.

Table 41: Transfer to small metal complexes. Experimental transition metal complexes that Janet and Kulik extracted from the CSD for which the base estimators agree and the maximum prediction probability is greater than .75.

CSD code	metal	ground truth	predicted oxidation state
EYUSO1	Mn	III	II
ECADOB	Co	II	II
EYUNIW	Co	II	II
DMAZCO	Co	II	II
ETUSOC02	Fe	II	II
DOQRAC	Fe	II	II
EXEHUM	Co	II	II
FALVEU02	Co	II	II
ECUGIS	Co	II	II
FEISXC01	Fe	II	II
EDETIT	Co	II	II
EZIROU	Co	II	II
ECOWEZ	Fe	III	III
EHEWIZ	Mn	II	II
ECODIM	Fe	II	II
DELVAS	Co	II	II
EYETUY01	Cr	III	III
FAQZEF	Co	II	II
CETDAG	Mn	II	II
EJEVEV	Mn	III	II
DUCJOA01	Co	II	II
ETUCED	Co	II	II

Continued on next page

Table 41: Transfer to small metal complexes. Experimental transition metal complexes that Janet and Kulik extracted from the CSD for which the base estimators agree and the maximum prediction probability is greater than .75.

CSD code	metal	ground truth	predicted oxidation state
EPASIY	Mn	II	II
EYOMUC	Co	II	II
DEPMOD	Mn	II	II
FAHLEI	Mn	III	III
DUCDIP	Fe	II	II
DUCBIN	Co	II	II
FAMBOL	Mn	II	II
EBUSEB	Co	II	II
EBIKOP	Co	II	II
ELAHII	Co	II	II
EXOWOD	Fe	II	II
DOXBEX	Co	II	II
FEHPYO	Fe	II	II
ABIWEO	Fe	II	II
EZOYFY	Mn	II	II
DELNIS	Co	II	II
DIRGES	Co	II	II
DIVFUL	Mn	II	II
CETJIU	Co	II	II
DOZROA	Fe	II	II
FATJIT	Co	II	II
DEVCUF	Co	II	II
ABORIU	Cr	III	III
DEGVET	Mn	II	II

Table 42: Transfer to metal complexes for predictions with low uncertainty. Global performance statistic for the prediction on 49 transition metal complexes assembled by Janet and Kulik for which all base estimators agree and for which the maximum prediction probability is greater than .75.

metric	value
accuracy macro	0.96 ± 0.03
F_1 macro	0.89
F_1 micro	0.96
κ^{593}	0.78 ± 0.15
$\kappa_{\text{no prevalence}}^{601}$	0.91
relative classifier information ⁶⁰²	0.55
precision macro	0.98
precision micro	0.96

Table 43: Transfer to metal complexes. Class statistic for the prediction on 63 transition metal complexes assembled by Janet and Kulik for which all base estimators agree and for which the maximum prediction probability is greater than .75.

metric	II	III
accuracy	0.96	0.96
adjusted F score	0.95	0.83
AUC	0.83	0.83
AUPR	0.98	0.83
precision	0.95	1.0
Matthews correlation coefficient ⁶⁰³	0.80	0.80
Gini index ⁶⁰⁴	0.67	0.67

D.5.6 Transfer to binary (ionic) solids

We retrieved all 590 stable (zero above complex hull on Perdew-Burke-Ernzerhof (PBE) functional level of theory) binary solids composed of Ag, Al, Au, B, Ba, Be, Bi, Ca, Cd, Ce, Co, Cr, Cs, Cu, Dy, Er, Eu, Fe, Ga, Gd, Hf, Hg, Ho, In, Ir, K, La, Li, Mg, Mn, Mo, Na, Nb, Ni, Os, Pb, Pd, Pr, Pt, Pu, Rb, Re, Rh, Ru, Sc, Sn, Sr, Ta, Tb, Tc, Ti, Tl, Tm, U, V, W, Y, Zn, Zr as metallic species and F, Cl, Br, I, O, S, N as anionic species from the Materials Project²²² using their API. Using these simple binary cases, we could use formal counting rules to automatically assign oxidation sites. The code to do so is available in the `oximachine_featurizer` package. Note that this test case is particularly harsh as we did not train our model with such solids and only had few examples for some of the metals and chemical environments.

Notably, we observe—as discussed in Section D.2.11—that the base estimators usually disagree in case of wrong predictions. For example, this filtering excludes cases VII and VIII, which we did not observe in training. Considering only the predictions in which the model is confident, we observe the performance statistics summarized in Tables 44 and 45. If we also consider the cases in which the model is not confident in its predictions, we find the statistics summarized in Tables 46 and 47. Overall, we observe that removing the cases in which the model is not confident significantly improves the scores.

The bad performance for the oxidation state IV appears to be due to lanthanides, mainly occurring in oxidation state III in MOF chemistry.

Generally, this analysis indicates that our approach can be extended to all chemistries.

Table 44: Transfer to binary solids for predictions with low uncertainty. Global performance statistic for the prediction on 189 binary (ionic) solids from the Materials Project for which all base estimators agree and for which the maximum prediction probability is greater than .75.

metric	value
accuracy macro	0.90
F ₁ macro	0.70
F ₁ micro	0.74
κ^{593}	0.64 ± 0.04
$\kappa_{\text{no prevalence}}^{601}$	0.48
relative classifier information ⁶⁰²	0.51
precision macro	0.78
precision micro	0.74

Table 45: Transfer to binary solids. Class statistic for the prediction on 189 binary (ionic) solids from the Materials Project for which all base estimators agree and for which the maximum prediction probability is greater than .75.

metric	I	II	III	IV	VI
accuracy	0.99	0.86	0.80	0.85	0.98
adjusted F score	0.99	0.81	0.84	0.56	0.69
AUC	0.99	0.82	0.82	0.63	0.71
AUPR	0.97	0.76	0.75	0.40	0.71
precision	0.94	0.80	0.67	0.47	1.0
Matthews correlation coefficient ⁶⁰³	0.96	0.67	0.60	0.31	0.65
Gini index ⁶⁰⁴	0.99	0.64	0.63	0.26	0.43

Table 46: Transfer to binary solids. Global performance statistic for the prediction on 590 binary (ionic) solids from the Materials Project.

metric	value
accuracy macro	0.90 ± 0.02
F_1 macro	0.36
F_1 micro	0.59
κ ⁵⁹³	0.44 ± 0.05
$\kappa_{\text{no prevalence}}$ ⁶⁰¹	0.05
relative classifier information ⁶⁰²	0.24
precision macro	None
precision micro	0.59

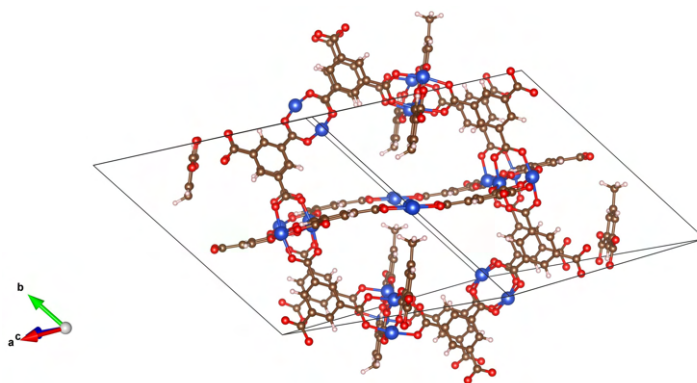
Table 47: Transfer to binary solids. Class statistic for the prediction on 590 binary (ionic) solids from the Materials Project.

metric	I	II	III	IV	V	VI	VII	VIII
accuracy	0.95	0.79	0.74	0.79	0.95	0.95	0.99	0.99
adjusted F score	0.84	0.74	0.78	0.57	0.29	0.48	0	0
AUC	0.84	0.74	0.75	0.62	0.53	0.60	0.50	0.50
AUPR	0.77	0.62	0.68	0.39	0.37	0.40	None	None
precision	0.86	0.59	0.59	0.41	0.67	0.60	None	None
Matthews correlation coefficient ⁶⁰³	0.75	0.47	0.48	0.26	0.20	0.33	None	None
Gini index ⁶⁰⁴	0.68	0.48	0.50	0.24	0.07	0.20	0.0	0.0

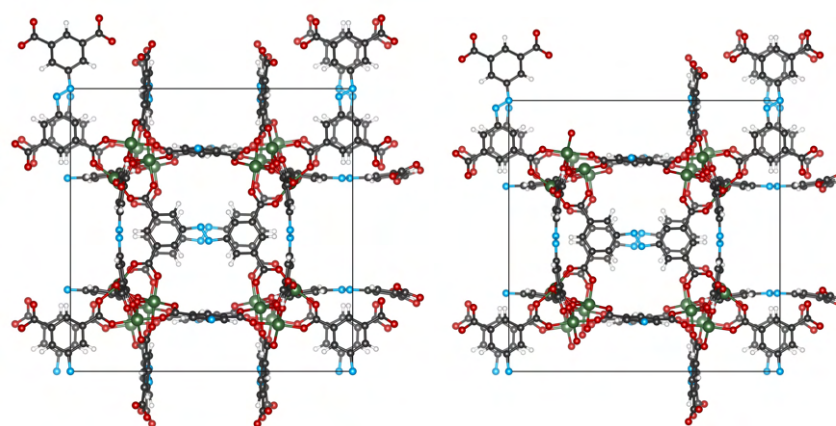
D.5.7 Case studies on defective MOFs

Given that our method only uses the local environment around a metal center, we were interested in how well it can deal with cases in which we distort the bonding of some linkers in well-known MOFs, which approximates missing-linker defects. Simply removing a linker will obviously lead to a smaller change in the local environment than in a real system, where the full electron density around a metal center will reorganize to relax into a new minimum—as for example, Smolders et al. reported for the transitions between Ce(III) and Ce(IV).⁶⁶³

As a first case study, we replaced one carboxylic acid group in some benzene-1,3,5-tricarboxylate linkers of Cu-BTC with hydrogen and methyl, respectively. The resulting structure is shown in Figure 137. Our model predicts Cu(I) with high confidence for sites that no longer bind to carboxyl groups (as required by charge balance).

**Figure 137: Defective Cu-BTC.** Simulating missing linkers with uncompensated charges by terminating some carboxyl groups with methyl and hydrogen, respectively.

Furthermore, linker defects do not necessarily need to change the oxidation state



(a) PCN-250 without axial water molecules (lost in the experiment below 150 °C). (b) PCN-250 without axial water molecules and a missing linker.

Figure 138: Thermally induced decarboxylation for PCN-250.

of the metal, as it is known for UiO-66 where different forms of charge compensation, e.g., with formate^{664–666} or hydroxy anions.^{667,668} We verified that our method predicts the correct oxidation state for all structures reported by Trickett et al.⁶⁶⁷ as well as all DFT optimized structures by Svane et al.⁶⁶⁹

An interesting example might also be the case of PCN-250 (Figure 138). Here, Zhou and co-workers reported an oxidation state change, from a Fe(III)_3 cluster to a Fe(II)Fe(III)_2 cluster, during activation of the framework, which was confirmed with ^{57}Fe Mössbauer spectroscopy.⁶⁷⁰ TGA-MS analysis revealed that an initial water loss is likely not responsible for the reduction of the Fe sites. Hence, a decarboxylation mechanism was proposed.

Similarly, our model does not predict oxidation state changes if we remove the axial water molecules. But, if we remove one linker, we observe a change in the oxidation state. We performed this analysis based on the crystal structure measured by Zhou and co-workers for the activated sample.

D.5.8 Case studies on charged frameworks and removal of charge-compensating counter ions

We also specifically investigated charged ligands or charged frameworks where we removed the counter ions. We picked those cases randomly from the CoRE-MOF structures with charged or ion annotation and made sure that the oxidation state was not in the CSD such that there was no way for our model have seen the structure with the correct oxidation state before.

These examples illustrate that due to the locality of our model, we do not depend on the direct knowledge of counter ions.

- Thapa et al. reported a charge-separated diamondoid MOF with anionic tetrahedral borate ligand and cationic Cu(I) metal ion. We correctly predict the +I oxidation state of the copper centers with high confidence.⁶⁷¹
- Xu et al. reported a $[\text{AMI}]_2[\text{Co}_3(\text{BDC})_4]$ MOF with $[\text{AMI}]^+$ (1-amy13-methylimidazolium) cations. We correctly predict Co(II) with and without the counter ions.⁶⁷²
- Nättinen and Rissanen reported a MOF with nitrate counter ions. Also after removal of those ions we correctly predict Cu(II) with high confidence⁶⁷³

- Eubank reported a MOF with nitrate counter ions. Also after removal of those ions we correctly predict Cu(II) with high confidence.⁶⁷⁴
- Huo et al. reported polyoxometalate-MOF hybrid in which channels are occupied by Keggin polyoxoanions. Also after removing the polyoxoanions we correctly predict the oxidation state for the Cu(I) sites.⁶⁷⁵
- Song et al. reported a polyoxometalate-MOF hybrid. Also after the removal of this ion we correctly predict Cu(II). Note that this structure (UCOCUM) has an oxidation state annotation in the CSD but was not used in our model due to timeout of the featurization for the large counter ion.⁶⁷⁶
- Bai et al. reported coordination polymers for which we correctly predict Ag(I) after removing the maleate.⁶⁷⁷
- With FARHOX Li et al. reported a Pb(II) MOF with cesium cations. Also after the removal of the cesium cations, we correctly predict Pb(II).⁶⁷⁸
- Van Albada et al. reported a mixed valent coordination polymer for which we also correctly predict Cu(I) and Cu(II) after removing the perchlorate counter ion.⁶⁷⁹

D.5.9 Case studies on low-dimensional frameworks and clusters

In addition, we investigated some examples of lower-dimensional MOF structures and clusters in more detail, showcasing that an approach that uses only the local environment of the metal center is transferable between different compound classes.

- Chaudhari et al. reported a π -stacked array of 2D-MOFs for which we correctly predict Zn(II) with high confidence.⁶⁸⁰
- Brown et al. reported a 1D-coordination polymer for which we correctly predict Cu(II) with high confidence.⁶⁸¹
- Park et al. reported a 2D semiconductive MOF for which we correctly predict Cu(II).⁶⁸²
- Wang et al. reported an inorganic-organic hybrid constructed from Keggin-type polyanions and copper clusters in which 1D bands are extended into a two-dimensional (2D) network. We correctly predict Cu(II) and W(VI).⁶⁸³
- Yamabayashi et al. reported the formation of a 3D framework from quantum dots. We predict the Ti(IV) and V(IV) oxidation states of the OD titanyl and vanadyl precursors correctly.⁶⁸⁴
- Tong et al. reported homo- and mixed valent $[2 \times 2]$ Co_4 Grid Complexes. For ILEPOG we correctly predict homovalent Co(II) (with intermediate confidence), and for ILEQAT, we correctly mixed valent Co(II/III).⁶⁸⁵
- Birk et al. reported Cr-F-Ln clusters for which we correctly predict Cr(III) and Ln(III).⁶⁸⁶
- Wong et al. reported Ru_2Co_2 $[2 \times \times 2]$ grids for which we correctly predict Ru(II) and Co(II).⁶⁸⁷
- Shiga et al. reported spin-crossover Fe(II) grids (e.g., MIQLEG) for which we correctly predict Fe(II).⁶⁸⁸

D.5.10 Case studies on solid-solution and bimetallic MOFs

- Fei et al.⁶⁸⁹ reported a series of solid solution cationic metal–organic materials with varying ratios of Co(II) and Zn(II). To demonstrate the robustness of our tool and how it can be applied to solid solution MOFs, we used the Transformer Python library⁶⁹⁰ to create a set of solid solutions with varying Co/Zn ratios. For this, we created the full set (6 unique structures of $\text{Co}_{4-x}\text{Zn}_x$) of solid solutions for a $1 \times 1 \times 2$ supercell and find Co(II) and Zn(II) in all cases. One could use a larger supercell for different Co/Zn ratios.
- A more challenging case might be a series of MOFs of the general formula $[\text{Al}(\text{OH})_{1-x}(\text{VO})_x\text{L}]_n$ (MIL-53/MIL-47) for which Kozachuk et al.³³⁸ probed the metal centers using EPR. Again, we used the Transformer library to create all unique solid solution structures (34 unique structures of $\text{Al}_{8-x}\text{V}_x$) of a $1 \times 1 \times 2$ supercell of MIL-53. We find an oxidation state of +III for the ideal solid-solution structures for all metal sites. In experiment, the authors found that the EPR silent V(III) ions are incorporated in the as synthesized samples but undergo oxidation upon activation. To reflect this in our solid solutions, we now created substituted structures starting from the activated structure of MIL-47. Our model consistently predicts Al(III) and V(IV).

D.5.11 Case study on robustness with respect to incorrect protonation

One might argue that upon oxidation of as-synthesized MIL-47(V) the bridges are $\mu\text{-O}$ instead of $\mu\text{-OH}$ groups, and that model might rely on this information. To test how sensitive the model is with respect to the protonation, we created a set of structures with varying protonation (considering $\mu\text{-O}$, $\mu\text{-OH}$ and $\mu\text{-H}_2\text{O}$ as bridging group, leaving all other atoms and the O itself fixed) and used them as input for our model. In all cases, we find V(IV) for the activated structure and V(III) for the as-synthesized one, supporting the discussion in the main text.

D.5.12 Case study on robustness with respect to the bond lengths

To analyze how sensitive our methods are w.r.t to changes in the bond lengths, we focused on the 190 structures from the CoRE-density-derived electrostatic charge (DDEC) database for which the oxidation state has been deposited in the CSD. In the CoRE-DDEC database, the structures have been relaxed using PBE, which tends to underbind for which reason pymatgen recommends scaling the bond lengths to be compatible with the bond valence parameters determined by O’Keefe and Brese⁶⁹¹ (https://pymatgen.org/pymatgen.analysis.bond_valence.html).

If we now compare the predictions of our model for the structures from the DFT optimized database with the ones for the experimental counterparts, we can estimate how sensitive the method is w.r.t. to changes in the bond lengths.

For all structures except one in this set, we find the same predictions for the experimental and the DFT optimized structure. The one exception is the Cu(II) coordination polymer TARWAK where there has been a larger change in coordination geometry upon relaxation (contraction of bonds of more than 0.6 \AA). But, the model trained on all structures from the CSD (section D.8) is more robust here and predicts consistent results also for this case.

D.6 POSSIBLE ERRORS IN THE CAMBRIDGE CRYSTALLOGRAPHIC DATABASE IDENTIFIED BY MEANS OF THE MODEL

Table 48: Possible errors in the CSD. Possible errors in the CSD assignment identified using the model: cases in which the assignment in the CSD disagrees with the assignment in the paper.

CSD reference code	metal center CSD	prediction
ACITAK ⁶⁹²	Cu(II)	Cu(I)
AFENAA ⁶⁹³	Cu(II)	Cu(I)
AQUCOF ⁶⁹⁴	Cu(I)	Cu(I/II)
AQUDAS ⁶⁹⁴	Cu(I)	Cu(I/II)
ATAGOR ⁶⁹⁵	Cu(II)	Cu(I)
BAXLAP ⁶⁹⁶	Cu(II)	Cu(I)
BICPOV ⁶⁹⁷	Cu(I)	Cu(II)
BIYWAK ⁶⁹⁸	Cu(II)	Cu(I)
COFYOM ⁵⁵⁹	Er(II)	Er(III)
COXQOV ⁶⁹⁹	Mn(II)	Mn(III)
CUVJUA ⁷⁰⁰	Cu(II)	Cu(I)
FIGZIG ⁷⁰¹	Cu(II)	Cu(I)
EBUPUO ⁷⁰²	Cu(II)	Cu(I)
EKIPAP01 ⁷⁰³	Cu(II)	Cu(I)
ENATUJ ⁷⁰⁴	Ga(II)	Ga(III)
EQIZAF ⁷⁰⁵	Cu(II)	Cu(I)
FIGTUL ⁷⁰⁶	U(III)	U(IV)
FIGZIG ⁷⁰¹	Cu(I)	Cu(II)
FIGZOM ⁷⁰¹	Cu(II)	Cu(I)
FIGZUS ⁷⁰¹	Cu(II)	Cu(I)
FODLAM02 ⁷⁰⁷	Cu(II)	Cu(I)
FUZXBU ⁷⁰⁸	Cu(II)	Cu(I)
GEVPOM ⁷⁰⁹	Hg(II)	Hg(I)
HAVHAQ ⁷¹⁰	Cu(II)	Cu(I)
HAWVIN ⁷¹¹	Co(II)	Co(III)
HAWVUZ ⁷¹¹	Co(II)	Co(III)
HITQOR ⁷¹²	Cu(II)	Cu(I)
HOJHUM ⁷¹³	Cu(II)	Cu(I/II)
HOMQE1 ⁷¹⁴	Cu(I)	Cu(II)
JAPYEH ⁷¹⁵	Cu(II)	Cu(I)
JAPYIL ⁷¹⁵	Cu(II)	Cu(I)
JAPYOR ⁷¹⁵	Cu(II)	Cu(I)
JAPYUX ⁷¹⁵	Cu(II)	Cu(I)
JAPZAE ⁷¹⁵	Cu(II)	Cu(I)
JAPZE1 ⁷¹⁵	Cu(II)	Cu(I)
ToDo: JAPZIM ⁷¹⁵	Cu(II)	Cu(I)
JAPZUY ⁷¹⁵	Cu(II)	Cu(I)
KAKTIA01 ⁷¹⁶	Cu(II)	Cu(I)
KAWFUL ⁷¹⁷	Cu(II)	Cu(I)
KESSIM ⁷¹⁸	Mn(II)	Mn(III)
KOGNIE ⁷¹⁹	Cu(II)	Cu(I/II)
KUTNUI ⁷²⁰	Cu(II)	Cu(II/I)
LARRAA ⁷²¹	Cu(II)	Cu(I)
LIDWAX ⁷²²	Cu(II)	Cu(I)
MIFQOJ/MIFQEZ ⁷²³	Mn(III)	Mn(II)
MOHQOQ ⁷²⁴	In(II)	In(III)
NAMTON ⁷²⁵	Cu(II)	Cu(I)
NECBUT ⁷⁰⁷	Cu(II)	Cu(I)
NECCUU ⁷²⁶	Cu(II)	Cu(I)
NOJSUD ⁷²⁷	Cu(I)	Cu(II)
NUZXAI ⁷²⁸	W(IV)	W(V)

Continued on next page

Table 48: Possible errors in the CSD. Possible errors in the CSD assignment identified using the model: cases in which the assignment in the CSD disagrees with the assignment in the paper.

CSD reference code	metal center CSD	prediction
OJANES ⁷²⁹	Hg(II)	Hg(I)
PEPVEM ⁷³⁰	Cu(II)	Cu(I)
PURVIH ⁷³¹	Cu(II)	Cu(I)
REKPUV ⁷³²	Cu(I)	Cu(II)
SIBDUD ⁷³³	Cu(II)	Cu(I)
SISMAL ⁷³⁴	Cu(II)	Cu(I)
SIYXIH ⁷³⁵	Fe(II)	Fe(III)
TAZGEG ⁷³⁶	Cu(II)	Cu(I)
TCAZCO ⁷³⁷	Co(III)	Co(II)
TEDDUC ⁷³⁸	Cu(II)	Cu(I)
TICMEY ⁷³⁹	Fe(II)	Fe(III)
UDOQAF ⁷⁴⁰	V(IV)	V(V)
UGARID/UGARID01 ⁷⁴⁰	Mo(II)	Mo(VI)
VUNQUS ⁵⁶⁰	Sm(II)	Sm(III)
XAHREE ⁷⁴¹	Cu(II)	Cu(II/I)
XEHVOW ⁷⁴²	Fe(II)	Fe(III)
YAKFIZ ⁷⁴³	Cu(II)	Cu(I)
ZASTUK ⁷⁴⁴	Mn(IV)	Mn(II)
RAXBAV ⁷⁴⁵	Fe(III)	Fe(II/III)
BAYMOF ⁷⁴⁶	Fe(II)	Fe(III)
GIMBIN ⁷⁴⁷	Mn(III)	Mn(II)
KIKPOM ⁷⁴⁸	Cr(II)	Cr(III)
JESMOJ ⁷⁴⁹	Hg(II)	Hg(I)

For most of the cases (47) listed in Table 48 we contacted the corresponding authors (for some of the structures, we could not trace down the e-mail address of the corresponding author). 14 (ca. 30%) responded and unequivocally confirmed the typo. One of the responses we received hints that the origin of the error can be traced back to the automatic naming in refinement programs.

To understand the limitations of our models better, we investigated misclassified cases from the holdout and training sets in more detail and, by that means, found errors in the assignment in the CSD which we list in Table 48.

Table 49: Unclear assignments. Examples of cases in which the model and paper disagree without further evidence for either assignment. Note that those cases are also counted as wrong predictions when we determine the classification metrics.

CSD reference code	metal centre CSD	prediction
AQONAW ⁷⁵⁰	Mn(III)	Mn(II/III)
KAKTIA01	Cu(II)	Cu(I)
PEHWEE	Co(III)	Co(II)
PEPVAI ⁷³⁰	Cu(II)	Cu(I)
QEWBAV	Fe(II)	Fe(III)
SESNAG ⁷⁵¹	Co(III)	Co(II)
SOXZIP	Co(III)	Co(II)
WUDLIQ ⁷⁵²	Sn(II)	Sn(IV)

From the confusion matrix, we can estimate that there are 1344 such metal sites in the test set where our model prediction disagrees with the assignment in the CSD. In 723 cases, this happens even though our model has high confidence in its predictions. In Table 50 we list some examples for which our model failed to predict the oxidation state.

Naturally, due to the limited amount of data, our model will tend to fail for rare oxidation states such as Re(I), Er(II), Yb(II), and Os(II).

Table 50: Potentially wrong predictions. Examples of potentially wrong predictions made by our model. Note that in several instances, there is no strong experimental support for the assignment.

CSD reference code	metal centre CSD/paper	prediction	note
AFAHAR ⁷⁵³	Re(I)	Re(III)	bad coverage in training set, one base estimator predicts Re(I)
ALICII ⁷⁵⁴	Cu(II)	Cu(I)	linear copper coordination typical for Cu(I)
FANSET ⁷⁵⁵	Ni(II/III)	Ni(III)	is unusual, there is no example for it in the training set
HURRER ⁷⁵⁶	Ce(IV)	Ce(III)	most similar structures from training set are Ce(III)
KAWQOP ⁷⁵⁷	Fe(II)	Fe(III)	most similar structures from training set are Fe(III)
UFILUR ⁷⁵⁸	Mn(II)	Mn(III)	model is not confident, predicts mixture of Mn(II) and Mn(III)
ZOQLEW ⁷⁵⁹	Pt(IV)	Pt(II)	metal site in training set is Pt(II) in WEJRIN
WUKNEV ⁵⁸⁰	Er(II)	Er(III)	rare oxidation state
SELJID ⁵⁶³	Eu(II)	Eu(III)	rare oxidation state
MAKWUS ⁵⁶²	Eu(II)	Eu(III)	rare oxidation state

D.7 BOND VALENCE SUM BASELINE

Given that the most widely used to assign oxidation states in crystalline materials is the BV sum method, we compared the performance of the BV sum method with our method for the case of copper. We chose copper as it is the most frequent metal in the CSD. Furthermore, this is also the metal Shields *et al.* focused on in their study, which includes the BV reparametrization for this element and its bonds with C, O, P, S, Cl, As, Se, Br, I, and 1-, 2- and 3-coordinated N.³⁰⁴

We found that approaches that try to self-consistently assign the oxidation state for all atoms in a structure, as implemented in *pymatgen*, are not converging in most cases. Moreover, they are unreliable in case of incorrect protonation. Therefore, we implemented the BV protocol to only assign the oxidation states of the metals (copper in this analysis), considering their environment of close atoms. Using the parameters of Shields *et al.*,³⁰⁴ we were able to reproduce most of the numerical results they reported: for only three values of over 24 reported in the paper, we see a deviation of more than 0.1 in the total bond valence sum (FOMHEU Cu(I), FAHHIC Cu(II) and BUACUM Cu(II)). The source of this deviation is unclear and possibly related to the convention used to define bonds. The code for our implementation is written in Python and available on GitHub as part of the *manage_crystal* package (https://github.com/danieleongari/manage_crystal). The final oxidation state is chosen as the one that has the minimum difference from the computed BV sum.

Note that, to use the BV protocol, we had to take into consideration a number of

caveats:

- Only bonds between Cu and C, O, P, S, Cl, As, Se, Br, I, and N were considered: bonds between Cu and other elements were simply discarded.
- All the atoms that are present in the CIF file were considered. Also the ones having a partial occupation.
- To adopt a consistent definition for the size of the atoms, the van der Waals diameter was defined as in the Universal Force Field (UFF).²¹⁷
- Copper was assumed to bond its neighboring atom when their distance is less than 80 % of the sum of their van der Waals radii.
- Nitrogen atoms need to be labeled as 1-, 2- or 3-coordinated, where this number corresponds to the *coordination number of donor atom in ligand, i.e., ignoring bonds to metal atoms*.³⁰⁴ Covalent bonds with nitrogens were assumed for atoms closer than 50 % of the sum of their van der Waals radii.
- In the case of nitrogen atoms with no bond, the parameter for general Cu-N bond was used. This lone nitrogen often indicates a bound amine solvent whose geometry is only partially resolved.
- Also in the case of nitrogen atoms with more than four bonds, we used the parameter for the general Cu-N bond. This is typically the case of disorders or partial occupations in the ligand that resolve in multiple disordered atoms.

The confusion matrix for the BV method, as applied for the labeled Cu-MOFs of the CSD, is reported in Table 51. Note that, for Cu(I) and Cu(II), 3.3 % and 17.9 % of the oxidation states are assigned incorrectly, respectively.

Table 51: Confusion matrix for the assignment using the bond valence sum method. Confusion matrix for the assignment of oxidation states for copper MOFs using the BV sum method.

		BV assignment	
		Cu(I)	Cu(II)
chemical name	Cu(I)	2370	81
	Cu(II)	1457	6687

D.8 TRAINING A MODEL ON ALL STRUCTURES OF THE CSD

Using the same approach we used for the MOF subset, we parsed the oxidation states for all (more than 1 million) compounds of the CSD. We attempted featurization for 354761 structures for which the oxidation state was reported in a form that can be used with our model, i.e., excluding mixed-valence compounds. Additionally, we parsed the COD for oxidation states. 879 CIFs, mostly inorganic solids, contain chemical names, saved under the `_chemical_name_systematic` tag, containing the oxidation state.

For 208,410 structures from the CSD and 811 from the COD the featurization succeeded, and we trained our model on those (applying a train/test split of 0.75/0.25 to allow for test set for a reasonable coverage of the chemical space). Note that the range of oxidation states in this set is larger (ranging from -1 to 8) than the one for

the MOF subset, and there is also no constraint to only a subclass of compounds, e.g., only polymeric compounds. Note that the coverage for oxidation states -1 (three examples) and 8 (seven examples) is still limited in this dataset. But, since the training set is now substantially larger, the predictive performance improves for metals and oxidation states rare in the MOF subset (and for which a model trained only on MOFs might not perform better than random guessing).

As for the model trained only on MOFs, we excluded those structures for which we found typos (see Table 53) and which are used as special test cases, e.g., due to experimental evidence, from the training set.

For an illustration of this dataset, we show in Figure 139 some representative structures and in Figure 140 the distribution of features and labels.

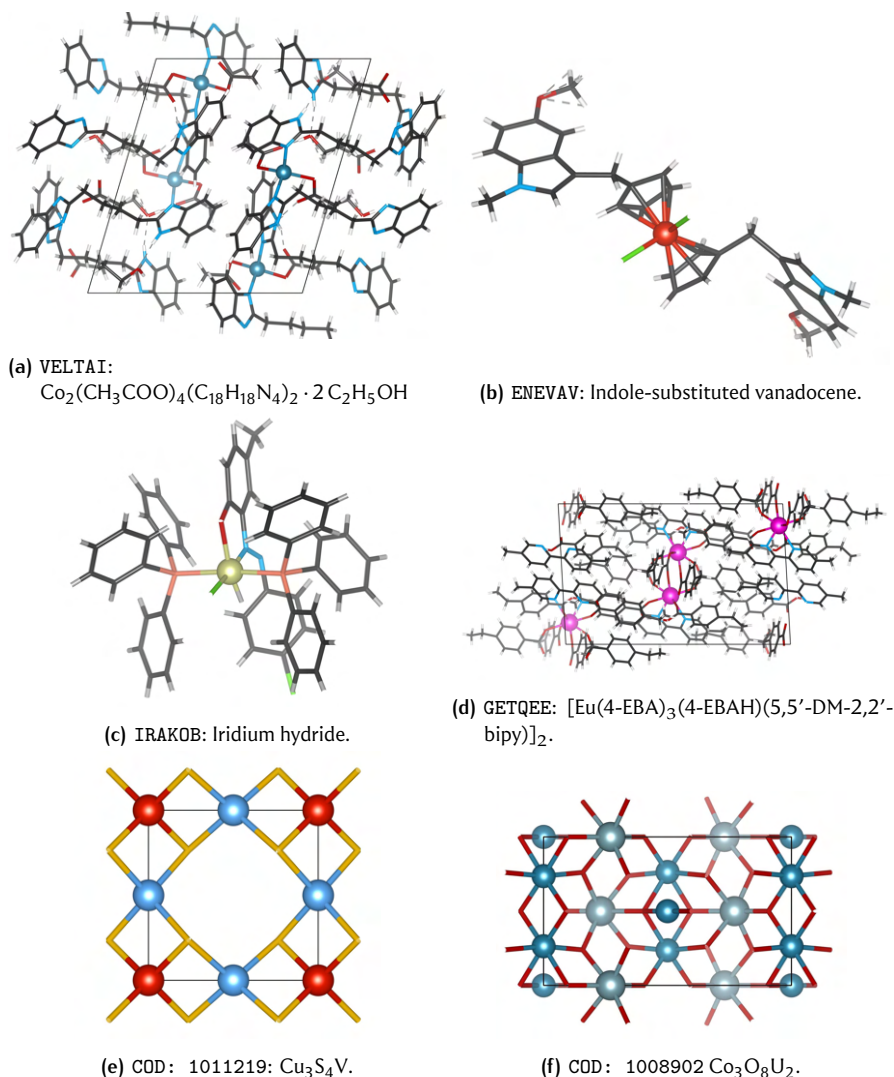


Figure 139: Structures from the dataset including all CSD and the COD. Structures were chosen using submodular selection in feature space.

D.8.1 Model architecture

We used the same model architecture as for the model that was only trained on structures from the MOF subset of the CSD. We also employed iterative stratification and dropped all duplicates (here meaning metal centers of the same metal with the same

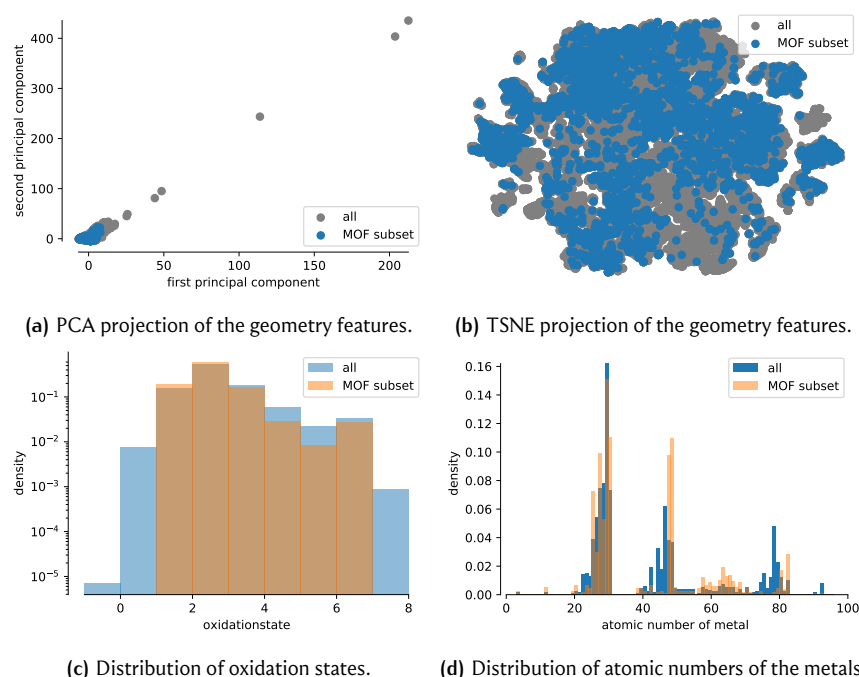


Figure 140: Coverage of the MOF dataset compared to the dataset sampled from all the CSD and COD with respect to the different feature and label scopes. Note that coverage is only one dimension of diversity quantification. One also needs to consider balance and disparity.¹³²

oxidation state in the same structure, which is a tighter criterion than the one we used for the model trained on the MOF subset). For hyperparameter optimization, we also employed a mixed search strategy with hyperopt, using the macro average of the F_1 score (i.e., placing a higher penalty on incorrect prediction of rarer oxidation states) on a validation set (that was also selected with iterative stratification) as the metric.

d.8.2 Potential typos identified in this analysis

Also with this model, we investigated 85 high-confidence predictions on the training set that disagree with the oxidation state reported in the chemical name in the CSD. Note that we could not access the original articles for all 85 structures.

Table 52: Possible errors in the CSD. Possible errors in the CSD assignment identified using the model: Cases in which the assignment in the CSD disagrees with the assignment in the paper. Found from 85 high-confidence predictions on the training set.

CSD reference code	metal center	CSD	paper
VAHCEM ⁷⁶⁰	Pd	III	II
ZUQVUC ⁷⁶¹	Co	II	III
CAVXAB ⁷⁶²	Mn	III	II
ZUVVUH ⁷⁶³	Ru	III	II
BERXAX ⁷⁶⁴	Co	III	II
BIWRUX ⁷⁶⁵	Cu	II	I
ZICXOY ⁷⁶⁶	Co	II	III
XUDJUB ⁷⁶⁷	Cd	I	II

Continued on next page

Table 52: Possible errors in the CSD. Possible errors in the CSD assignment identified using the model: Cases in which the assignment in the CSD disagrees with the assignment in the paper. Found from 85 high-confidence predictions on the training set.

CSD reference code	metal center	CSD	paper
ZASBEC ⁷⁶⁸	Co	II	III (valence transition close to RT)
PEXRIU ⁷⁶⁹	Cu	II	I
TACJUE ⁷⁷⁰	Yb	II	III
MATVEK ⁷⁷¹	Co	II	III
YUSKIH ⁷⁷²	Ru	III	II
SEBCUA ⁷⁷³	Cu	I	I/II
NAYCIZ ⁷⁷⁴	Ru	I	III
XAVJAF01 ⁷⁷⁵	Cu	II	I
YAZPOG ⁷⁷⁶	Co	II	III
AQOPEB ⁷⁷⁷	Co	III	II
BAQMIR ⁷⁷⁸	Pt	I	II
BIPLIX ⁷⁷⁹	Co	II	III
ZEGFOG10 ⁷⁸⁰	Mo	III	V
KIFVOL ⁷⁸¹	Cu	I	II
CAGROT ⁷⁸²	Co	II	III
ROFDEV ⁷⁸³	Au	II	I
JIVWUH ⁷⁸⁴	Pd	III	II
BIYWEO ⁶⁹⁸	Cu	II	I
AKIYUO ⁷⁸⁵	Cu	III	II
ROFCUK ⁷⁸³	Au	II	I

Table 53: Possible errors in the CSD. Possible errors in the CSD assignment identified using the model: Cases in which the assignment in the CSD disagrees with the assignment in the paper. Found in a less systematic analysis of predictions on the training and test set.

CSD reference code	metal center	CSD	paper
PEQXUF ⁷⁸⁶	Au	I	III
YEJJUT ⁷⁸⁷	Au	III	I
ROFDAR ⁷⁸³	Au	II	I
BARGAF ⁷⁸⁸	Cu	II	I
ZIPFOT ⁷⁸⁹	Cu	II	I
WALWEM ⁷⁹⁰	Cu	II	I
UCIDEQ ⁷⁹¹	Pt	I	II
VIGQUA ⁷⁹²	Pt	O	II
ZOCLAE ⁷⁹³	Au	II	I
FELHUZ10 ⁷⁹⁴	Cu	II	I
QICXEG ⁷⁹⁵	Cu	I	II
WUBLEK01 ⁷⁹⁶	Cu	II	I
ZERHAF ⁷⁹⁷	Cu	II	I
FARROH ⁷⁹⁸	Cu	II	I
ZEJXAQ ⁵⁴⁷	Ho	II	III
UHOGUT ⁷⁹⁹	Sn	IV	II/IV
XALXAJ ⁸⁰⁰	Cu	II	I
DEFCID ⁸⁰¹	Cu	II	I
YEGLAZ ⁸⁰²	Mo	IV	VI
ZEJWIX ⁵⁴⁷	Er	II	III
CUNTOV ⁸⁰³	Co	II	III
BAWGEO ⁸⁰⁴	Cu	II	I
BUPHEA ⁸⁰⁵	Cu	II	I
OLELAS ⁸⁰⁶	Cu	I	II
DOCPOD ⁸⁰⁷	Co	II	III
WOKJOW ⁸⁰⁸	Zr	III	IV
WOQKET ⁵⁵⁵	Co	I	II
WOKJAI ⁸⁰⁸	Zr	III	IV
BEFZAP ⁸⁰⁹	Cu	II	I
VUQWAG ⁸¹⁰	W	V	IV

Continued on next page

Table 53: Possible errors in the CSD. Possible errors in the CSD assignment identified using the model: Cases in which the assignment in the CSD disagrees with the assignment in the paper. Found in a less systematic analysis of predictions on the training and test set.

CSD reference code	metal center	CSD	paper
GATDAH ⁸¹¹	Cu	II	I
WEZPEX ⁸¹²	Cu	I	II
LEKFEM ⁸¹³	Pt	I	II
KIWDOJ ⁸¹⁴	Re	VII	V

The potential typos we find with our model indicate that our model could have utility in verifying oxidation state assignments, also for chemistries that are more complex than the one of MOFs.

A fascinating case from this experiment might be the one of [Ru(bpy)₃]. Echegoyen and co-workers proposed a Ru(0) center to explain the lack of an EPR signal and the lack of counter ions.⁸¹⁵ The corresponding structure is deposited as TIWPEU in the CSD. Our model predicts Ru(II), and follow-up studies⁸¹⁶ suggested that the ligand rather than the metal is reduced, supporting the notion of a Ru(II) center, which our model predicts.

d.8.3 Performance analysis

As for the model trained on the MOF subset, we analyzed the performance on different subsets of the test set with various metrics. The overall confusion matrix is shown in Figure 141, overall metrics are summarized in Table 54, statistics per elements are reported in Table 55 and confusion matrices only for elements with a minimum variance in oxidation states are shown in Figure 142–145.

As for the model trained and tested on the MOF subset of the CSD, we find good predictive performance for all oxidation states—especially for the confident predictions.

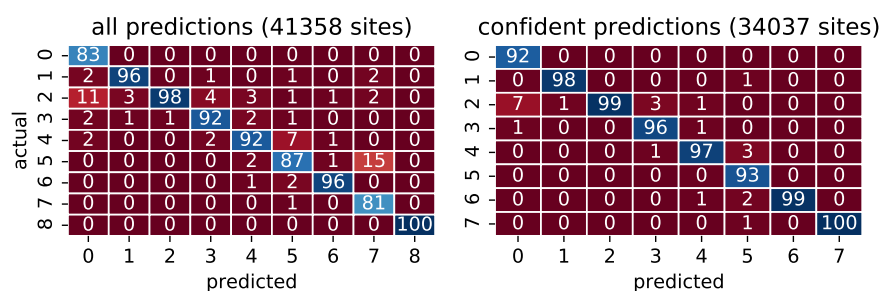


Figure 141: Confusion matrix for the model trained and tested on CSD and COD.

For the AUC the average metrics in Table 54 are provided as

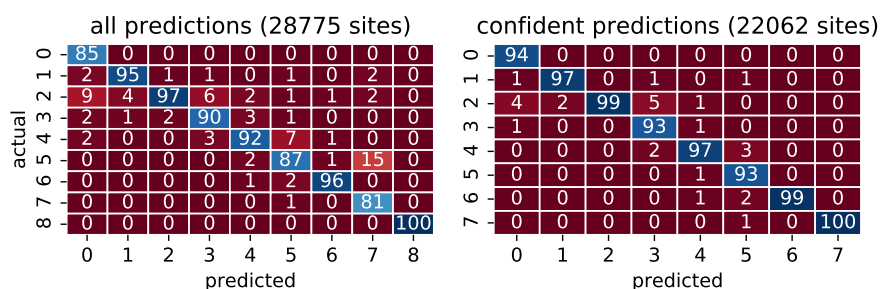
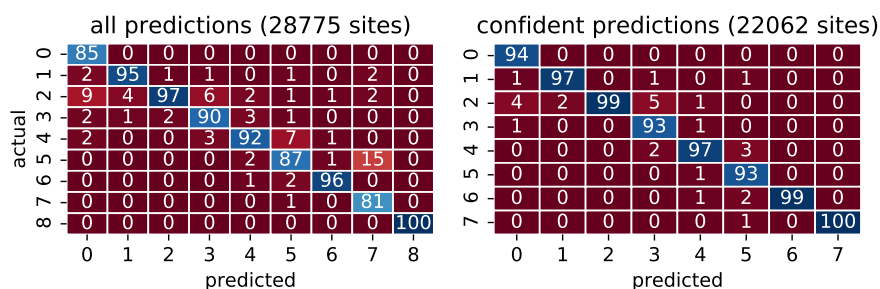
$$\text{AUNU} = \frac{\sum_{i=1}^{|C|} \text{AUC}_i}{|C|} \quad (33)$$

and the version weighted with the class proportions

$$\text{AUNP} = \sum_{i=1}^{|C|} \frac{P_i}{\text{POP}} \text{AUC}_i. \quad (34)$$

Table 54: Performance metrics for the model trained and tested on CSD and COD. Decimals for values greater 0.99 are cut and not rounded.

metric	value (all predictions)	value (only confident predictions)
accuracy	0.99	0.99
ROC AUNU	0.96	0.98
ROC AUNP	0.97	0.99
F ₁ micro	0.96	0.98
F ₁ macro	0.92	0.96
precision micro	0.96	0.98
precision macro	0.91	0.95
recall micro	0.96	0.98
recall macro	0.92	0.97
MCC	0.93	0.97
κ	0.93	0.97

**Figure 142:** Confusion matrix for only elements for which there are at least two oxidation states for which the least frequent one occurs with at least 2 % frequency in the MOF subset of the CSD.**Figure 143:** Confusion matrix for only elements for which there are at least two oxidation states for which the least frequent one occurs with at least 5 % frequency in the MOF subset of the CSD.**Table 55: Classification metrics on the test set as a function of the metal.** Micro and macro refer to the averaging methods, where micro averaging gives high weights to rare classes.

metal	accuracy	recall (micro)	precision (micro)	recall (macro)	precision (macro)
Ag	0.999	0.999	0.999	0.833	1.0
Am	1.0	1.0	1.0	1.0	1.0
Au	0.994	0.994	0.994	0.542	0.665
Ba	1.0	1.0	1.0	1.0	1.0
Be	1.0	1.0	1.0	1.0	1.0
Bi	1.0	1.0	1.0	1.0	1.0

Continued on next page

Table 55: Classification metrics on the test set as a function of the metal. Micro and macro refer to the averaging methods, where micro averaging gives high weights to rare classes.

metal	accuracy	recall (micro)	precision (micro)	recall (macro)	precision (macro)
Ca	1.0	1.0	1.0	1.0	1.0
Cd	1.0	1.0	1.0	1.0	1.0
Ce	0.977	0.977	0.977	0.97	0.955
Co	0.979	0.979	0.979	0.613	0.715
Cr	0.959	0.959	0.959	0.885	0.985
Cs	1.0	1.0	1.0	1.0	1.0
Cu	0.989	0.989	0.989	0.658	0.657
Dy	0.997	0.997	0.997	0.5	0.498
Er	1.0	1.0	1.0	1.0	1.0
Eu	0.991	0.991	0.991	0.625	0.995
Fe	0.938	0.938	0.938	0.428	0.435
Ga	0.975	0.975	0.975	0.5	0.494
Gd	0.993	0.993	0.993	0.5	0.497
Hf	0.981	0.981	0.981	0.5	0.491
Hg	0.996	0.996	0.996	0.333	0.332
Ho	1.0	1.0	1.0	1.0	1.0
In	0.994	0.994	0.994	0.875	0.997
Ir	0.957	0.957	0.957	0.467	0.473
K	1.0	1.0	1.0	1.0	1.0
La	1.0	1.0	1.0	1.0	1.0
Li	1.0	1.0	1.0	1.0	1.0
Lu	1.0	1.0	1.0	1.0	1.0
Mg	0.991	0.991	0.991	0.5	0.496
Mn	0.982	0.982	0.982	0.703	0.774
Mo	0.941	0.941	0.941	0.814	0.92
Na	1.0	1.0	1.0	1.0	1.0
Nb	0.955	0.955	0.955	0.8	0.783
Nd	0.996	0.996	0.996	0.5	0.498
Ni	0.987	0.987	0.987	0.391	0.557
Np	0.889	0.889	0.889	0.875	0.917
Os	0.919	0.919	0.919	0.692	0.734
Pb	1.0	1.0	1.0	1.0	1.0
Pd	0.991	0.991	0.991	0.333	0.373
Pr	1.0	1.0	1.0	1.0	1.0
Pt	0.992	0.992	0.992	0.55	0.798
Pu	1.0	1.0	1.0	1.0	1.0
Rb	1.0	1.0	1.0	1.0	1.0
Re	0.947	0.947	0.947	0.863	0.936
Rh	0.943	0.943	0.943	0.549	0.572
Ru	0.951	0.951	0.951	0.52	0.767
Sc	0.957	0.957	0.957	0.5	0.479
Sm	0.981	0.981	0.981	0.75	0.991
Sn	0.984	0.984	0.984	0.95	0.991
Sr	1.0	1.0	1.0	1.0	1.0
Ta	0.875	0.875	0.875	0.705	0.856
Tb	0.988	0.988	0.988	0.5	0.494
Tc	0.935	0.935	0.935	0.656	0.74
Th	1.0	1.0	1.0	1.0	1.0
Ti	0.933	0.933	0.933	0.444	0.644
Tl	1.0	1.0	1.0	1.0	1.0
Tm	0.949	0.949	0.949	0.819	0.819
U	0.954	0.954	0.954	0.778	0.91
V	0.955	0.955	0.955	0.603	0.779
W	0.936	0.936	0.936	0.59	0.766
Y	1.0	1.0	1.0	1.0	1.0
Yb	0.994	0.994	0.994	0.997	0.978
Zn	1.0	1.0	1.0	1.0	1.0
Zr	0.981	0.981	0.981	0.333	0.327

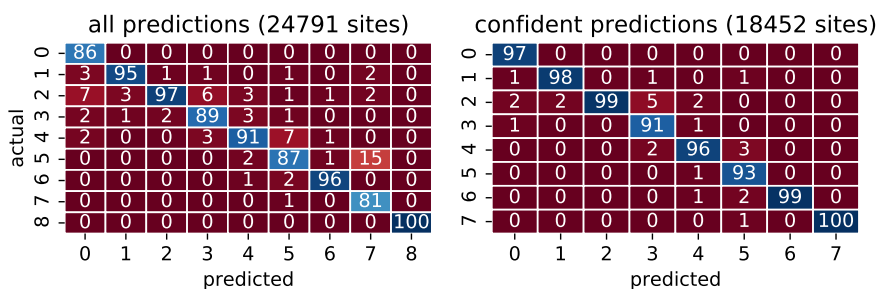


Figure 144: Confusion matrix for only elements for which there are at least two oxidation states for which the least frequent one occurs with at least 10 % frequency in the MOF subset of the CSD.

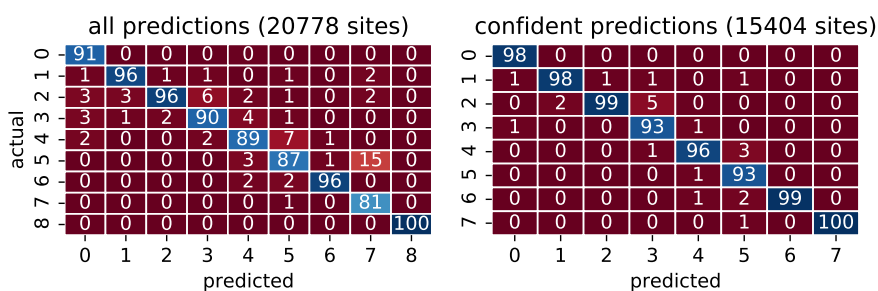


Figure 145: Confusion matrix for only elements for which there are at least two oxidation states for which the least frequent one occurs with at least 20 % frequency in the MOF subset of the CSD.

D.8.4 Future work

For future work, we also plan to include all structures from the Inorganic Crystal Structure Database (ICSD), i.e., improve the coverage for purely inorganic compounds, and then use a neural network with Monte-Carlo sampling as uncertainty estimate (to increase the efficiency of the model with larger datasets). Potentially, rarer oxidation states, e.g., Fe(I), Fe(IV), Fe(V), could be oversampled by creating hypothetical compounds.

E

SUPPORTING INFORMATION FOR “MACHINE LEARNING FOR INDUSTRIAL PROCESSES: FORECASTING AMINE EMISSIONS FROM A CARBON CAPTURE PLANT”

E.1 INTERMITTENCY SCENARIOS

Due to the electricity grid load following requirements and the increasing penetration of intermittent renewables, fossil fuel-based electricity generation can play an important role in load balancing the grids. In such a flexible operation, it is important that the capture unit can follow these dynamics. Most of the studies so far focused on technical and economic challenges associated with the flexible operation of the capture plant.^{356,379,380} However, a key aspect when assessing the implications of power plant flexibility is how the CO₂ capture plant needs to be operated to comply with future environmental legislation with respect to amine emissions.

Our study focused on intermittency scenarios that could have a strong effect on amine emissions. As a baseline, we assume the capture plant operates with the power plant at full load. Therefore, each scenario looks at the different consequences of reducing the load of the power plant on a daily basis and hence expecting a lower flow rate of flue gas entering the absorber column.

For the emissions, not only the reduced load of the flue gas is relevant, but at the same time, there can be other effects. For example, a possible consequence of a part-load operation can be a reduction in the steam flow entering the reboiler unit or other changes. By combining the possible consequences, our stress test consists of the following eight scenarios:

1. *Water wash temperature increase*; one of the possible side effects of fluctuations in the flue gas flow is that it can result in changes in the temperature in the absorber, which directly impacts the temperature of the water wash. Also, a sudden change in CO₂ concentration caused by unstable operation of the power plant can directly affect the water wash temperature.
2. *Water wash flow rate decrease*; increasing the water wash flow rate is most frequently used to control amine emissions.⁸¹⁷ It is, therefore, important to see if this control mechanism functions sub-optimally.
3. *Flue gas temperature increase*; a flue gas temperature variation might result from unstable plant operation.
4. *Lean solvent flow rate decrease*; one of the mechanisms to control variations in the capture rate is to change the solvent flow rate.⁸¹⁸
5. *Lean solvent and flue gas flow rate decrease*; a possible part-load operation of the power plant due to less electricity production.⁸¹⁹
6. *Lean solvent temperature increase*; a side effect of the variation of the steam supply to the reboiler can impact the solvent temperature.

7. *Increase of both the lean solvent and water wash temperatures*; one of the possible side effects of both variations in the steam and solvent flow rate and in the flue gas flow.
8. *Capture rate decrease*; is a side effect of a reduced steam availability that might happen when the power plant operates at part load and is expected when a higher amount of electricity supply to the grid is required, which can cause a drop in the capture rate.

Each scenario is translated into a change of different process variables that can cause the desired effect. For example, the water wash temperature is changed by changing the flow rate of the cooling water in the heat exchanger of the water wash section. The details are given in Table 56.

In this testing phase, no regime of self-accelerating degradation occurred, even if phases with increased reboiler temperature and increased oxygen concentration in the flue gas give reason to expect an increased degradation rate of CESAR1. Note that due to the design of the experimental campaign, we cannot directly capture changes that occur over a timescale longer than 24 h.

Table 56: Parametric tests with the CESAR1 solvent at post-combustion capture (PCC) pilot plant at Niederaussem. Experiments were performed as step changes and selected to cover relevant regions of interest in the space of operating conditions. The experiments and a preliminary analysis have already been described in Charalambous et al.³⁵⁸

exp.	region of interest				description
	WW	FG	LS	CRate	
1	■				Water wash temperature (TI-19) increase from $\approx 45^{\circ}\text{C}$ to 55°C ¹
2	■				Water wash flow (FI-19) decrease from 6000 kg h^{-1} to 5000 kg h^{-1} ²
3		■			Flue gas temperature (TI-3) increase from 45°C to 55°C ³
4			■		Lean solvent flow (FI-11) decrease from $\approx 2400\text{ kg h}^{-1}$ to 2000 kg h^{-1} ⁴
5		■	■		Lean solvent flow (FI-11) decrease from $\approx 2400\text{ kg h}^{-1}$ to 2000 kg h^{-1} and flue gas flow (FI-2) decrease from 1500 kg h^{-1} to 1300 kg h^{-1}
6		■	■		Lean solvent temperature (TI-12/TI-13) increase from 43°C to 52°C ⁵
7	■		■		Lean solvent temperature (TI-12/TI-13) increase from 43°C to 53°C and water wash temperature (TI-19) increase to 42°C
8				■	Capture rate decrease from 90 % to 80 % ⁷

¹ This is achieved by changing the temperature of the cooling medium (i.e., water) in the heat exchanger (heat exchanger (HEX)) located in the water wash section.

² This is achieved by changing the speed of the pump located in the water wash section.

³ This is achieved by changing the temperature of the cooling medium in the HEX located in the direct contact cooler (DCC) unit.

⁴ This is achieved by modifying the amount of the solvent in the pilot plant.

⁵ This is achieved by controlling the temperature of the cooling medium in the HEX located after the pump, which sends the lean solvent from the stripper to the absorber.

⁶ In this experiment the water wash baseline temperature was 33°C .

⁷ This is controlled by the steam flow in the reboiler and the reboiler level.

Abbreviation: WW, water wash; FG, flue gas; LS, lean solvent; CRate, CO_2 capture rate.

The actual experimental campaign included two additional experiments (see Appendix E.3), which we did not include in our analysis due to the plant's instability after a power plant shutdown.

E.2 EXPERIMENTAL METHODS

The plant is equipped with online gas-analysis systems to continuously monitor the composition of the inlet and outlet gas streams. The online gas-analysis systems are used to quantify the composition of (i) flue gas at the absorber inlet (i.e., CO₂, CO, NO: BA5000 Bühler infrared spectroscope; O₂: BA3500, Bühler, paramagnetic detection, SO₂: MCS 100E, Sick/Maihak, photometric detection limit <4 mg m⁻³), (ii) CO₂-lean flue gas downstream of the water wash outlet, and (iii) CO₂ product stream.

The GasMET analyzers used for monitoring solvent emissions exiting the water wash section have been calibrated for standard inorganic components (i.e., NH₃, SO₂, NO_x, CO, CO₂) and for AMP and Pz. The detection limit for amines is approx. 1 mg m⁻³ (STP). Both GasMET analyzers were zeroed with nitrogen at the beginning of the experimental campaign and re-zeroed once a week during the measuring period.

All liquid samples were analyzed by ATR-FTIR spectroscopy, for which two calibration sets were used. One calibration was performed for CO₂ loadings and amine concentrations within the range expected for lean and rich samples. The other calibration was performed for very low amine concentrations and was used to monitor the CO₂ loading and amine concentrations in the water wash. The detection limit for AMP and Pz are around 0.3 wt%.

E.3 EXPERIMENTAL DATA

Table 57 lists all dynamic tests and the steady-state performance of the pilot plant in terms of the CO₂ capture rate, CO₂ loadings, amine and CO₂ emissions, and re-boiler duty. Figure 146 illustrates the water wash performance comparing the water wash temperature and the emissions (i.e., CO₂, AMP, Pz, H₂O, NH₃) at the exit of the water wash section. The grey areas in the figure represent the times when no dynamic tests were performed (days 4–5 and 11–16).

Table 57: Summary of pilot plant emissions and pilot plant performance. The parametric tests are indicated with the symbol 'X.'

Run	Step-Change Parameter	L/G ratio	Loading / wt%			CO ₂ Capture Rate / %	Emissions / mg/m ³ ‡			Reboiler duty MJ/kg CO ₂
			Lean	Rich	WW		AMP	Pz	CO ₂ ‡	
1	Baseline	1.56	0.09	0.29	0.53	86.3	64.2	10.2	1.16	2.8
2	X Water wash (WW) temp.	1.56	0.07	0.30	0.42	86.5	152.1	19.7	1.18	4.2
	Average over 24h	1.56	0.08	0.30	0.45	87.0	102.7	24.9	1.22	3.2
3	Baseline	1.53	0.07	0.32	0.57	86.6	60.2	24.9	1.41	3.1
4	X WW flow rate	1.53	0.07	0.31	0.48	86.3	67.3	14.3	1.40	3.4
	Average over 24h	1.53	0.07	0.31	0.51	85.7	64.5	21.1	1.46	2.9
5	Baseline	1.53	0.09	0.31	0.58	84.1	63.2	18.8	1.43	3.0
6	X Flue gas (FG) temperature	1.53	0.09	0.34	0.52	83.8	65.3	12.9	1.49	3.1
	Average over 24h	1.53	0.09	0.30	0.56	85.4	72.2	23.0	1.35	2.7
7	Baseline	1.53	0.07	0.31	0.58	84.4	77.1	59.4	1.63	2.3
8	X Lean solvent (LS) flow rate	1.30	0.05	0.26	0.67	77.3	48.5	20.6	2.78	2.8
	Average over 24h	1.48	0.06	0.31	0.62	82.8	63.0	28.0	1.96	3.0
9	Baseline	1.53	0.07	0.28	0.54	84.6	34.8	20.6	1.90	3.2
10	X LS and FG flow rates	1.53	0.07	0.27	0.57	83.5	50.6	18.0	1.07	3.9
	Average over 24h	1.53	0.07	0.29	0.55	83.3	60.1	24.5	1.78	3.0
11	Baseline	1.53	0.07	0.28	0.50	85.0	56.8	16.0	1.11	4.2
12	X LS temperature	1.53	0.07	0.27	0.41	87.7	97.4	8.0	1.07	2.6
	Average over 24h	1.53	0.07	0.30	0.45	86.3	70.3	17.8	1.08	2.8
13	Baseline	1.53	0.08	0.27	0.53	81.3	62.6	23.0	1.47	2.6
14	X LS and WW temperature	1.52	0.07	0.28	0.39	84.2	126.6	3.5	1.25	2.7
	Average over 24h	1.53	0.07	0.29	0.44	85.5	82.0	15.8	1.15	3.0
15	Baseline	1.56	0.08	0.30	0.52	82.6	75.8	28.3	1.43	2.4
16	X Capture rate	1.36	0.07	0.30	0.64	75.5	51.1	23.2	2.25	2.4
	Average over 24h	1.51	0.07	0.28	0.60	83.2	60.2	20.8	1.46	3.3

† At standard temperature and pressure (STP).

‡ CO₂ is given in vol%.

Abbreviation: WW, water wash; FG, flue gas; LS, lean solvent.

E.3.1 Pilot plant parameters

The typical operational parameters and the boundary conditions of the capture plant are provided in Table 58. The average values of these parameters are also provided before the start of the dynamic campaign (on day 275 with operation with the CE-SAR1 solvent).

Table 58: Typical operational parameters and boundary conditions used during the testing campaign. Including average values before the start of the campaign.

Parameter	Unit	Value	Value †	Description
Flue gas temperature at DCC inlet	°C	64	63	—
Flue gas temperature at absorber inlet	°C	40–45	44	—
Flue gas flow rate	m ³ h ⁻¹ ‡	1150	1150	—
CO ₂ content of the flue gas	vol%, dry	12.5	12.5	Measured at absorber inlet
O ₂ content of the flue gas	vol%, dry	5.0	5.4	Measured after desulphurization
SO ₂ content of the flue gas	mg m ⁻³ ‡	<1.0	<1.0	Measured at absorber inlet
Dust	mg m ⁻³ ‡	<2.0	<2.0	—
NO _x content of flue gas	mg m ⁻³ ‡	100–160	100–160	Measured at absorber inlet
NO ₂ content of flue gas	mg m ⁻³ ‡	6–8	6–8	Measured at absorber inlet
Solvent flow rate	kg h ⁻¹	2600	2400	—
Water circulation in the DCC	kg h ⁻¹	8000	9758	—
pH value of water in the DCC	—	7–7.2	7–7.2	—
CO ₂ -lean flue gas temperature	°C	40–45	46	Measured at water wash outlet
Solvent regeneration temperature	°C	120	120	—
Desorber pressure	bar(a)	1.75	1.75	—
CO ₂ capture rate	—	90	92	—
Specific energy demand	GJ/tCO ₂	3.0	3.3	For solvent regeneration

† Average values measured before the start of the dynamic campaign (on day 275).

‡ Dry value at standard temperature and pressure (STP).

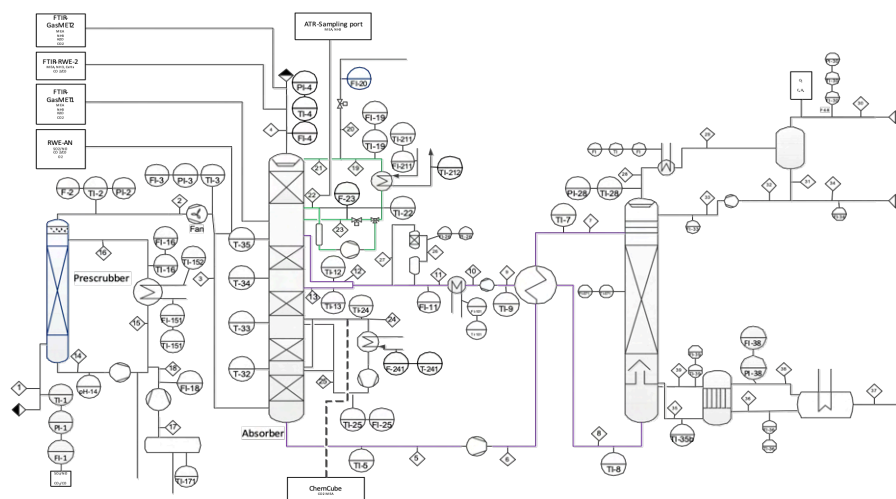


Figure 147: Detailed PID diagram. Abbreviations are explained in section E.3.2.

E.3.2 Abbreviations

- **PI-2** Pressure of the flue gas stream at the exit of the direct contact cooling (DCC) unit.
- **TI-2** Temperature of the flue gas stream at the exit of the DCC unit.
- **FI-2** Flow rate of the flue gas stream entering the absorber column.
- **PI-3** Pressure of the flue gas stream upstream of the absorber column.
- **TI-3** Temperature of the flue gas stream upstream of the absorber column.
- **CO2-3** CO₂ concentration of the flue gas upstream of the absorber column.
- **O2-3** O₂ concentration of the flue gas upstream the absorber column.
- **TI-32** Temperature at the top of the 1st bed (counting from the bottom of the absorber).
- **TI-33** Temperature at the top of the 2nd bed (counting from the bottom of the absorber).
- **TI-34** Temperature at the top of the 3rd bed (counting from the bottom of the absorber).
- **TI-35** Temperature at the top of the 4th bed (counting from the bottom of the absorber).
- **PI-4** Pressure of the treated gas exiting the water wash section.
- **TI-4** Temperature of the treated gas exiting the water wash section.
- **FI-4** Flow rate of the treated gas exiting the water wash section.
- **CO2-4** CO₂ concentration of the flue gas at the water wash exit – output (CO₂ emissions).
- **NH3-4** NH₃ concentration of the flue gas at the water wash exit – output (NH₃ emissions).

- **FI-11** Flow rate of the lean solvent entering the absorber (at the top of the column).
- **TI-12** Temperature of the lean solvent entering the absorber (at the top of the column).
- **TI-13** Temperature of the lean solvent entering the absorber.
- **FI-20** Flow rate of fresh water added to the water wash section.
- **FI-211** Flow rate of water in the HEX located at the water wash section.
- **TI-211** Temperature of water in the heat exchanger located at the water wash section.
- **TI-212** Temperature of water out of the HEX located at the water wash section.
- **TI-8** Temperature of the lean solvent leaving the desorber sump (upstream of the HEX).
- **TI-9** Temperature of the lean solvent downstream of the HEX.
- **TI-5** Temperature of the rich solvent exiting the absorber.
- **TI-7** Temperature of the rich solvent downstream of the HEX
- **TI-28** Temperature of the CO₂ concentrated stream exiting the top of the stripper.
- **PI-28** Pressure of the CO₂ concentrated stream exiting the top of the stripper.
- **PI-30** Pressure of the CO₂ concentrated stream exiting the condenser (after the stripper).
- **TI-30** Temperature of the CO₂ concentrated stream exiting the condenser.
- **FI-30** Flow rate of the CO₂ concentrated stream exiting the condenser.
- **FI-38** Flow rate of the steam entering the reboiler.
- **PI-38** Pressure of the steam entering the reboiler.
- **FI-36** Flow rate of the stream exiting the reboiler.
- **TI-36** Temperature of the steam exiting the reboiler.
- **Reb. Duty** Reboiler energy use.
- **FI-19** Flow rate of the water wash section (water wash circulation rate).
- **TI-19** Temperature of the water wash.
- **PI-1** Pressure of the flue gas upstream to the DCC unit.
- **TI-1** Temperature of the flue gas upstream to the DCC unit.
- **TI-35b** Temperature of the rich solvent into the reboiler.
- **FI-35** Flow rate of the solvent into the reboiler.
- **TI-39** Temperature of the solvent out of the reboiler.
- **FI-23** Flow rate of excess liquid leaving the water wash (WW) section into the absorber.

- **TI-22** Temperature of the water in the WW section upstream of the WW pump.
- **Level Des.** Level of the liquid into the desorber.
- **Level Reb.** Level of the liquid into the reboiler.
- **TI-24** Temperature of the liquid upstream to the HEX located in the intercooling section.
- **TI-25** Temperature of the liquid downstream to the HEX of the intercooling section.
- **FI-25** Flow rate of the liquid in the intercooling section.
- **FI-16** Flow rate of the liquid downstream of the HEX located in the DCC unit.
- **TI-16** Temperature of the liquid downstream of the HEX located in the DCC unit.
- **FI-151** Flow rate of the water in the HEX of the DCC unit.
- **TI-151** Temperature of the water in the HEX of the DCC unit.
- **TI-152** Temperature of the water out of the HEX of the DCC unit.
- **FI-241** Flow rate of the water in the HEX of the intercooling section.
- **TI-241** Temperature of the water in the HEX of the intercooling section.
- **TI-242** Temperature of the water out of the HEX of the intercooling section.

E.4 EXPLORATORY DATA ANALYSIS

Using the Augmented-Dickey Fuller test (as implemented in the statsmodels library³⁸⁴) we tested for stationarity and found that the raw data is non-stationary due to deterministic and non-deterministic trend components (see also autocorrelation functions in Figure 148). For the machine learning analysis, we removed the deterministic trend component using linear regression through the endpoints. We then tested for Granger causalities (Figure 149) using the statsmodels library.

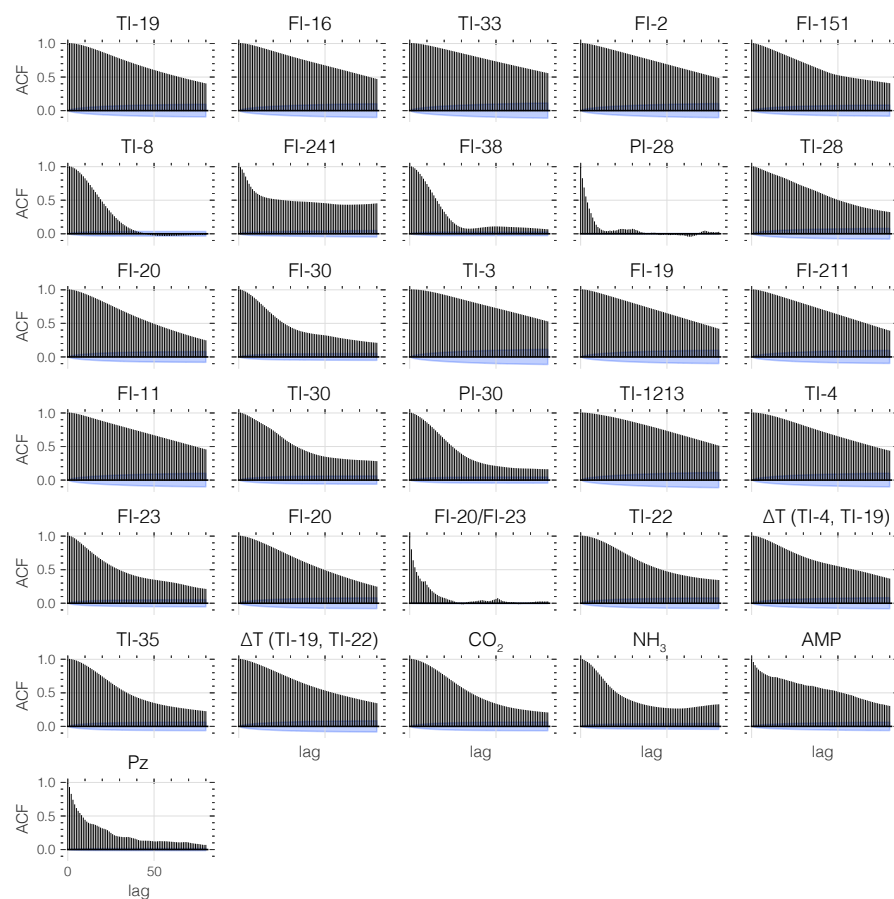


Figure 148: Autocorrelation functions (ACF). Calculated up to a maximum lag of 80 (i.e., 160 min). The blue region shows a .95 confidence interval.

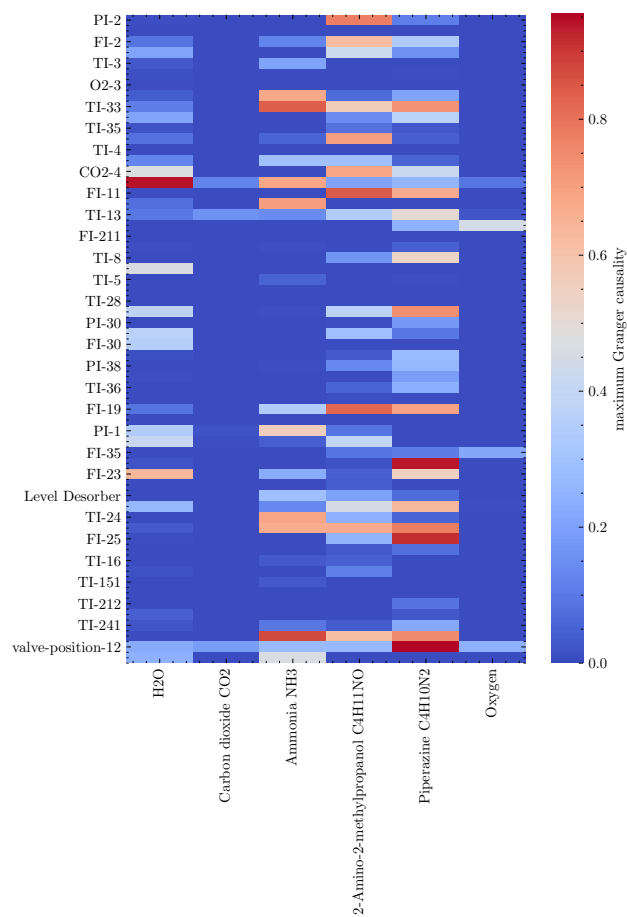


Figure 149: **Granger causalities.** Strongest Granger causalities between features and labels in the dataset, computed up to a lag of 20.

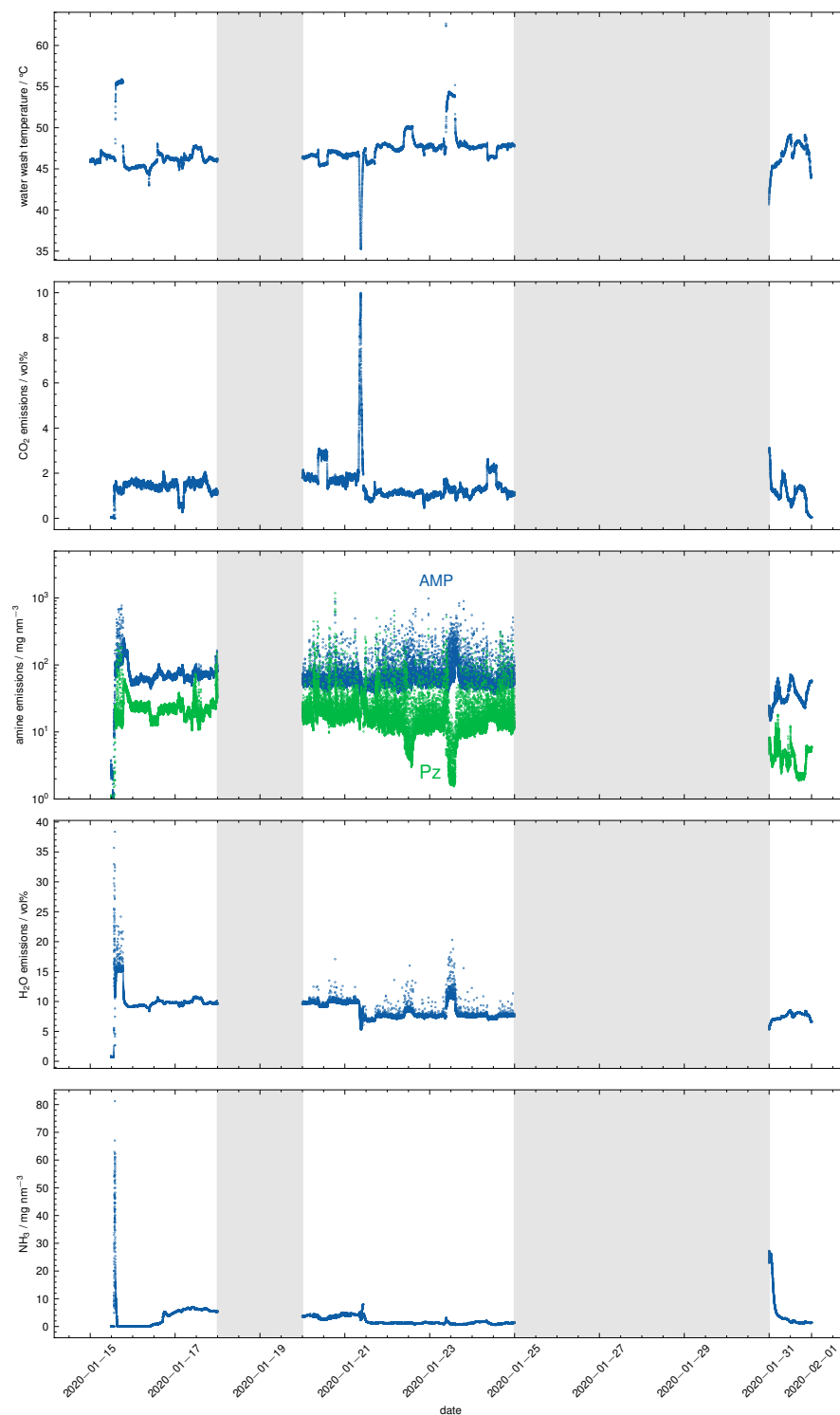


Figure 146: Water wash temperature and CO₂, AMP, Pz, H₂O vapor, and NH₃ emissions profiles as recorded using the GasMET FTIR at the water wash outlet during the CESAR1 dynamic campaign (i.e., 10 dynamic experiments over 10 operational days). The grey area denotes the period when no dynamic experiments were performed (i.e., days 4–5 and days 11–16).

E.5 DATA PRE-PROCESSING

We performed a range of pre-processing steps to make the data amenable to machine learning. Figure 150 illustrates their impact on the time series. One can observe that we preserve the shape of the time series but reduce the intensity of the spikes (but we still preserve their presence and location).

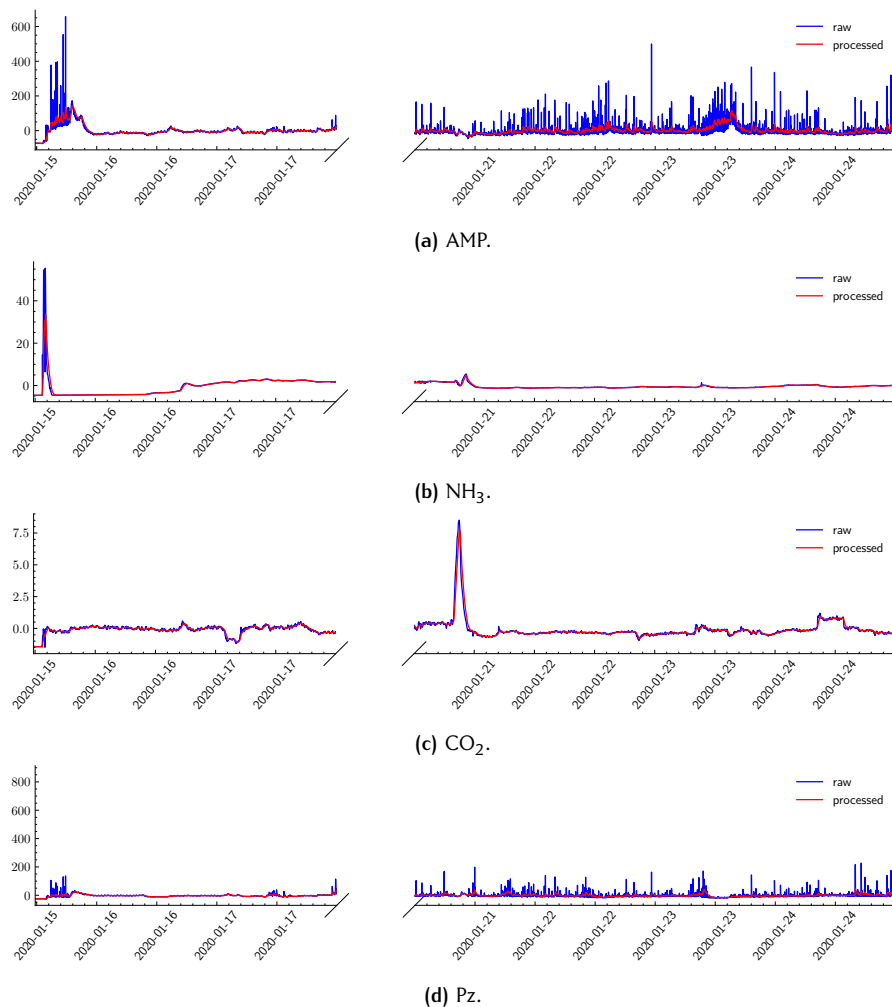


Figure 150: Impact of data pre-processing. The subfigures show impact on different objectives.

E.6 FEATURE SELECTION

Feature selection was guided by domain knowledge (i.e., excluding parameters expected to have no causal relation with the emissions) and aided by Granger causality heatmaps shown in Figure 149.

In the final models, we used the following feature set:

- TI-19: Temperature of the liquid in the water wash.
- FI-19: Flow rate of the liquid in the water wash.
- FI-11: Flow rate of the lean solvent entering the absorber.

- TI-3: Temperature of the flue gas entering the absorber.
- TI-12/TI-13: Temperature of the lean solvent entering the absorber
- TI-35: Temperature at the topmost bed of the absorber

For causal impact analysis, we also included (TI-35 - TI-4), i.e., the temperature difference of the temperature at the topmost bed of the absorber and the temperature of the treated gas exiting the water wash section

E.7 QUANTILE REGRESSION USING GRADIENT BOOSTED DECISION TREES

Gradient-boosted decision trees can be used for time series forecasting using a concatenated lagged time series as input. One new regressor can then be trained for every point in the forecasting horizon. Here, we use the LightGBM implementation.³⁸¹

To produce uncertainty intervals, we regress on the quantile loss, $l_\alpha(x)$, which for quantile α reads

$$l_\alpha(x) = \begin{cases} -(1 - \alpha)x & \text{if } x \leq 0 \\ \alpha x & \text{if } x \geq 0 \end{cases} \quad (35)$$

E.7.1 Performance

We use multiple metrics to measure the predictive performance of our models given two time series y^1 and y^2 of length T .

MEAN ABSOLUTE ERROR (MAE)

$$\text{MAE} = \frac{1}{T} \sum_{t=1}^T (|y_t^1 - y_t^2|). \quad (36)$$

MEAN ABSOLUTE PERCENTAGE ERROR (MAPE)

$$\text{MAPE} = 100 \cdot \frac{1}{T} \sum_{t=1}^T \left| \frac{y_t^1 - y_t^2}{y_t^1} \right|. \quad (37)$$

where y_t^1 is the actual time series (ground truth).

OVERALL PERCENTAGE ERROR (OPE)

$$\text{OPE} = 100 \cdot \left| \frac{\sum_{t=1}^T y_t^1 - \sum_{t=1}^T y_t^2}{\sum_{t=1}^T y_t^1} \right|. \quad (38)$$

where y_t^1 is the actual time series (ground truth).

Table 59: Performance metrics. Metrics for the median historical forecasts of the gradient-boosted decision tree model for different forecasting horizons.

	AMP			Pz		
	2 min	1 h	2 h	2 min	1 h	2 h
MAE / a.u.	0.0089	0.039	0.036	0.0095	0.050	0.040
MAPE / %	2.4	11	9.5	4.3	23	21
OPE / %	0.38	0.34	3.8	2.0	17	10

E.7.2 Hyperparameter Optimisation

For all models except the ones used for the causal impact analysis, we searched hyperparameters on the grid presented in Table 60. For the causal impact analysis, we did not optimize every feature lag separately but used the same lag for all covariates. We used the Bayesian optimization implemented in the weights and biases platform (<https://docs.wandb.ai/guides/sweeps>) for all searches. We only performed a hyperparameter search for the 0.5 quantiles to limit computational cost and reused the same hyperparameters for the other quantiles. However, we separately optimized hyperparameters for AMP and Pz. Also, for the first step change we had, due to the short window preceding the step change, limited the search to lags smaller than 40.

Table 60: Parameter ranges for hyperparameter search. Hyperparameter ranges are considered for the GBDT models.

parameter name	range
lag	uniform in (0, 200)
feature lag	uniform in (-200, 0)
n_estimators	uniform in (50, 1000)
bagging_freq	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
bagging_fraction	uniform in (0.001, 1.0)
num_leaves	integers in (1, 200)
extra_trees	(True, False)
max_depth	(-1, 10, 20, 40, 80, 160, 320)

Table 61: Hyperparameter settings. Hyperparameter settings for AMP for different output horizons.

parameter	2 min	60 min	120 min
bagging_fraction	0.572	0.989	0.948
bagging_freq	1	4	10
extra_trees	False	False	False
lag_1	-1	-158	-169
lag_2	-134	-70	-118
lag_3	-55	-196	-38
lag_4	-10	-125	-97
lag_5	-23	-102	-7
lag_6	-176	-113	-141
lags	100	70	174
max_depth	-1	20	40
n_estimators	743	375	559
num_leaves	7	177	74

Table 62: Hyperparameter settings. Hyperparameter settings for Pz for different output horizons.

parameter	2 min	60 min	120 min
bagging_fraction	0.331	0.121	0.180
bagging_freq	10	8	7
extra_trees	False	False	False
lag_1	-93	-136	-81
lag_2	-182	-144	-170
lag_3	-111	-146	-62
lag_4	-106	-114	-77
lag_5	-56	-55	-154
lag_6	-90	-83	-75
lags	100	171	195
max_depth	40	40	80
n_estimators	126	275	193
num_leaves	4	90	113

We used a forecasting horizon twice the step change's duration for the causal impact analysis.

Table 63: Hyperparameter settings. Hyperparameter settings for AMP for different step changes (causal impact analysis).

parameter	step 0	step 1	step 2	step 3	step 4	step 5	step 6
bagging_fraction	0.334	0.475	0.742	0.526	0.955	0.947	0.847
bagging_freq	10	8	3	5	10	0	2
extra_trees	True	False	True	False	False	True	False
covariates lag	-51	-172	-30	-71	-184	-23	-126
lags	47	5	145	167	194	173	121
max_depth	80	160	320	160	-1	80	160
n_estimators	123	866	816	348	880	394	511
num_leaves	40	178	177	42	162	167	130

Table 64: Hyperparameter settings. Hyperparameter settings for Pz for different step changes (causal impact analysis).

parameter	step 0	step 1	step 2	step 3	step 4	step 5	step 6
bagging_fraction	0.817	0.959	0.685	0.241	0.785	0.796	0.900
bagging_freq	9	2	10	3	0	2	10
extra_trees	True	True	True	True	False	True	False
covariates lag	-27	-10	-106	-19	-10	-79	-150
lags	13	185	170	46	77	10	130
max_depth	20	320	20	160	40	40	160
n_estimators	69	221	92	130	979	975	933
num_leaves	48	109	84	67	73	44	9

E.8 TEMPORAL CONVOLUTIONAL NEURAL NETWORKS

Figure 151 gives an overview of the modeling process using temporal convolutional neural networks.

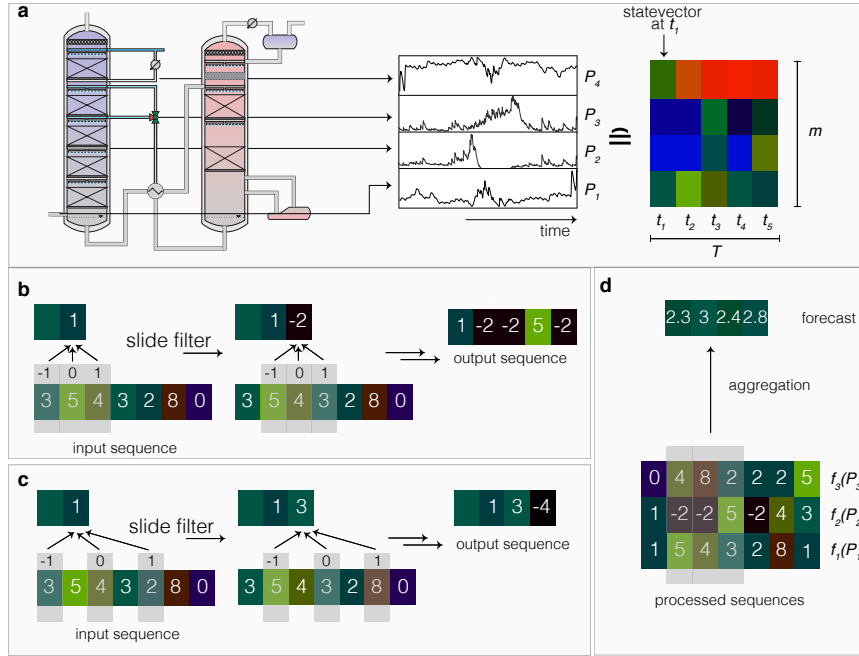


Figure 151: Schematic illustration of the modeling approach using temporal convolutional neural networks. Mapping of the data of the plant onto an “image”; the data set can be thought of an “image” with “width” equal to length of the input sequence (T) and “height” equal to the number of parameters, m . We represent with colors the value of parameter P_j at a time t_i . As the predictions should be invariant to the order of the rows, we only apply the pattern learning via convolutional filters (light grey) in the time direction. Therefore the image of the plant should be seen as m one-dimensional image. **b**, Convolutional kernels are slid over the m images as part of the pattern recognition algorithm. The weights of the kernels (of the f th filter), $W_f = [w_{-1}^f, w_0^f, w_1^f]$, are initially set according to a conventional initialization scheme and learned in the training procedure. In the first layer, the kernel operates on directly neighboring values. In order to allow for the model to learn different representations (patterns), we use multiple learnable filters per layer of the neural network,^{382,820} i.e., the layer outputs multiple (e.g., 64) one-dimensional “images”. The output of one layer is fed into the next layer as an input. **c**, To allow learning of correlations across large time scales, as they are expected to be relevant in industrial processes, we add “holes” to the kernels (dilated convolutions) that operate on the output of preceding convolutional layers. **d**, The results of all the kernel operations (after applying operations of the forms of **b** and **c** multiple times) are all collected via a “2D” convolution into a predicted emission. This schematic shows that our output sequence cannot be longer than T , the length of the input sequence. To deal with the “edges”, we apply (causal) padding with zeros at the front of the input sequence (not shown in the figure).

E.8.1 Model architecture

Temporal convolutional neural networks contain multiple key design elements (see Figure 152):

- *layers of dilated causal convolutions*: convolutional layers found widespread use in computer vision applications, and a key factor contributing to the success is the concept of weight sharing. In practice, it has been found that such models are much easier to train than recurrent neural networks. One problem with convolutional layers is that they usually only have a local receptive field. To remedy this problem, dilated convolutions have been developed which differ from “normal” convolutional kernels by having “holes” in the kernel. By increasing the size of the holes one can achieve exponential increases in the receptive field of the model. The term “causal” refers to the fact that one wants to avoid lookahead, i.e., a forecasting model should not depend on future data. For this reason, zero padding is only applied to one side of the time series.
- *residual connections*: also this technique was first developed in the context of computer vision,⁸²¹ and it was empirically found that the option to skip some layers via an “identity mapping” can stabilize training and boost predictive performance.
- *weight normalization*: is a reparametrization trick that was found to stabilize and speed up the convergence of training.
- *dropout*: is a well-known technique for regularizing neural networks that work by randomly disabling certain weights.

Note that this is not a model architecture we specifically design for this work. It is already available in the darts library.³⁸²

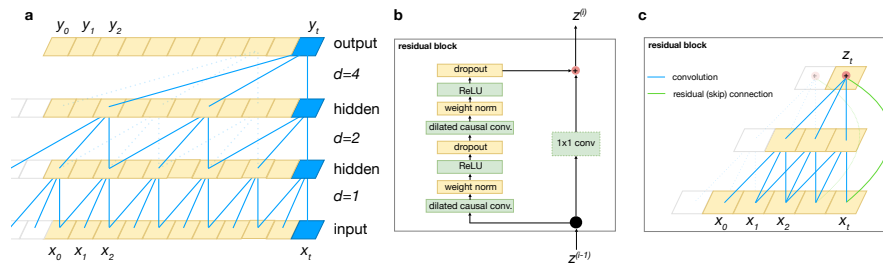


Figure 152: Building blocks of TCN models. Figure adapted from Bui et al.¹¹ **a** Dilated convolutions allow exponential increases in temporal resolutions via the use of convolutions with “holes”. The parameter d indicates the dilation rate. **b** A residual block contains layers of dilated causal convolutions, dropout, ReLU activation, and weight normalization. The network also has the possibility to “skip” the dilated convolution. **c** Dilated convolutions with skip connections form a block.

Models were trained on an NVIDIA Quadro RTX 6000 graphics card within a few minutes.

E.8.2 Monte Carlo Dropout

In statistics, one distinguishes between epistemic and aleatoric uncertainty. Aleatoric uncertainty captures noise inherent in the data, whereas epistemic uncertainty is

the uncertainty in the model (prior distribution over the weights). With the Monte-Carlo dropout approach, we approximate the latter term by approximating the sampling from the posterior by enabling the dropout layers during inference time. This approach has already been used, for example at Uber,⁸²² to estimate the uncertainty of forecasts. Note that we did not add a term for the aleatoric uncertainty in our uncertainty estimates. Also note that the theoretical justification of this uncertainty estimation procedure is still debated.^{823,824}

E.8.3 Hyperparameter optimisation

For hyperparameter optimization, we focused on predicting the amine emissions and optimized on a validation set of the AMP emissions. (Implicitly assuming that hyperparameters that perform well for AMP will also perform well for Pz.) For this, we performed a time-based split using the first 50 % of the data for training, the subsequent 25 % for validation, and the last 25 % for testing. Note that in our case this is a particularly challenging setting for the model as every day different intervention were performed. We considered the hyperparameter grid in Table S 65 and optimized for the mean absolute percentage error on historical forecasts on the validation set.

For hyperparameter optimization, we used the Bayesian optimization approach implemented in weights and biases (<https://docs.wandb.ai/guides/sweeps>).

Table 65: Hyperparameter grid considered in this work and final settings. The 60 min, 10 min, and 2 min output sequence length model reached a validation mean absolute percentage error of 10.4 %, 12.05 %, and 12.35 %, respectively.

parameter	range	120 min out-put	60 min out-put	10 min out-put	2 min output
number of convolutional layers	[4, 8, 16]	4	8	4	16
number of filters	[8, 16, 32, 64]	32	64	64	16
weight norm	true / false	false	true	false	false
kernel size	[2, 3, 4, 5]	2	4	3	3
dropout probability	uniform distribution between 0.1 and 0.9	0.5617	0.3668	0.3239	0.1511
batch size	[32, 64, 128]	64	64	128	128
number of epochs	[100, 200, 300, 400]	100	200	100	200
input sequence length	[31, 40, 60, 61, 80, 160] (timestamps)	61	80	31	80
learning rate	uniform sampling in logarithmic space between 10^{-5} and 10^{-1}	0.0192	0.0297	0.0100	0.01197

E.8.4 Forecasting performance

See Fig. 152 for historical forecasts.

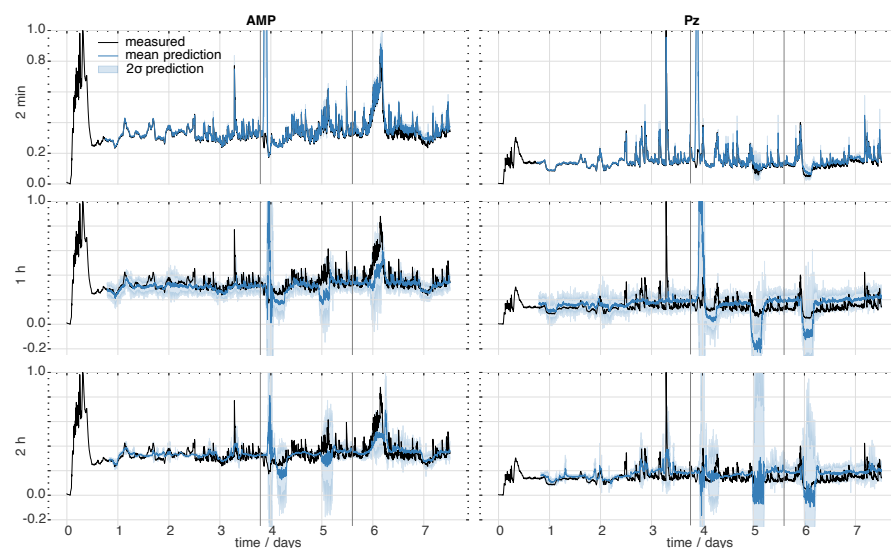


Figure 153: Historical forecasts for temporal convolutional neural networks for three output sequence lengths. The 60 min output sequence length model achieved an MAE of 12.05 % on the validation set. The 10 min output sequence length model achieved an MAE of 12.35 % on the validation set. The 2 min output sequence length model achieved an MAE of 12.35 % on the validation set. The vertical lines indicate the validation/test split points. The black curves show the measured emissions, and the blue curves show the forecasts with the solid line indicating the mean and the band indicating the 2σ interval estimated from 50 Monte-Carlo dropout runs.

E.9 VARIMA BASELINE

We also attempted to use a vector autoregressive moving average with exogenous regressors model (VARIMA) model ($p = q = 5$ with constant and linear trend term*) on the same covariates. The predictions are shown in Fig. S 154. We see that also a VARIMA model can learn from the data, however, it—as one would expect—struggles to predict the spikes.

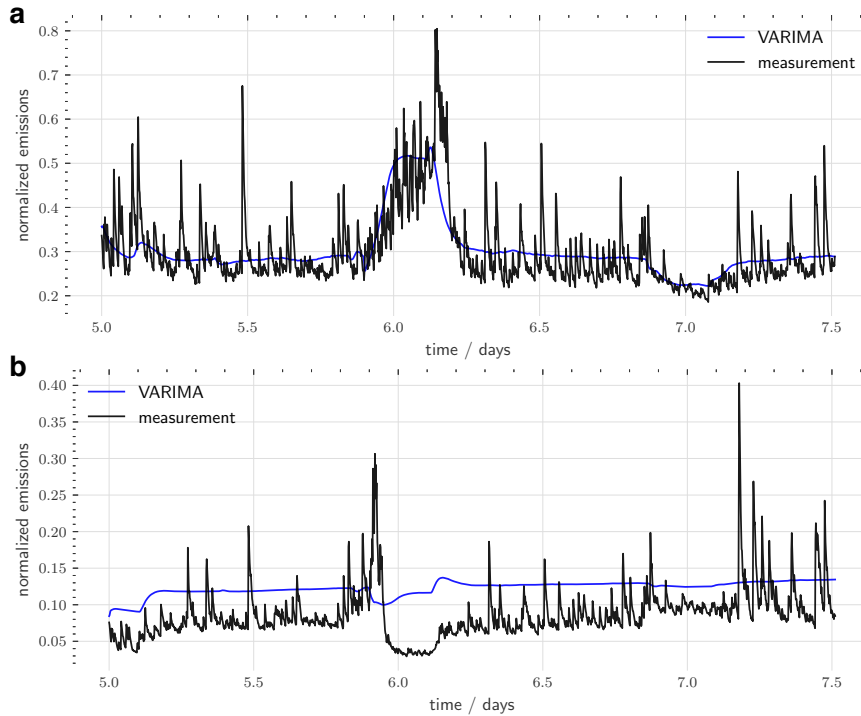


Figure 154: Forecasts vs. measured AMP (a) and Pz (b) emissions for a VARIMAX model.

Additionally, we investigated the use of the recently proposed Temporal Fusion Transformers.⁸²⁵ However, within our limited tuning (testing hidden layer sizes 8, 16, 32), we did not find them to outperform the temporal convolutional model (while being more expensive in training and inference).

E.10 CAUSAL IMPACT ANALYSIS

Table 66: Performance metrics. Model performance in the pre-intervention periods for AMP.

day	MAPE / %	OPE / %
1	200	nan
2	3.5 1.9	
3	4.4	1.3
4	14	15
5	8.9	3.4
6	9.2	10
7	18	19

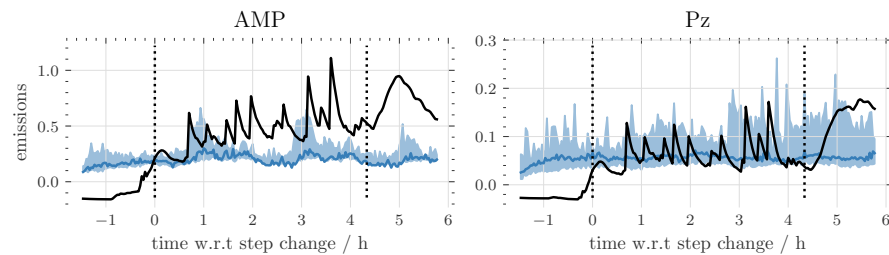
* The forecast quality for $p = 20, q = 0$, and $p = 10, q = 0$ are comparable to the ones shown here.

Table 67: Performance metrics. Model performance in the pre-intervention periods for Pz.

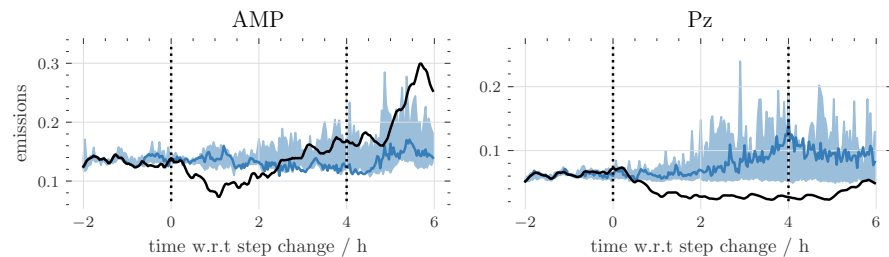
day	MAPE / %	OPE / %
1	670	nan
2	9.2	9.4
3	8.6	6.9
4	46	52
5	32	36
6	14	16
7	32	36

E.10.1 Day 1: Step increase in water wash temperature

For both amines, we observe a significant increase in emissions. However, the model is unreliable for this step change as there is only little data preceding the step change.

**Figure 155: Causal impact analysis for day 1.** Causal impact analysis for the step increase in water wash temperature.**E.10.2 Day 2: Step decrease in water wash flow**

For both amines, we observe a decrease in emissions.

**Figure 156: Causal impact analysis for day 2.** Causal impact analysis for the step decrease in water wash flow rate.**E.10.3 Day 3: Step increase in flue gas temperature**

Even though the measured emissions (black) might suggest an increase in emissions, this increase is not statistically significant.

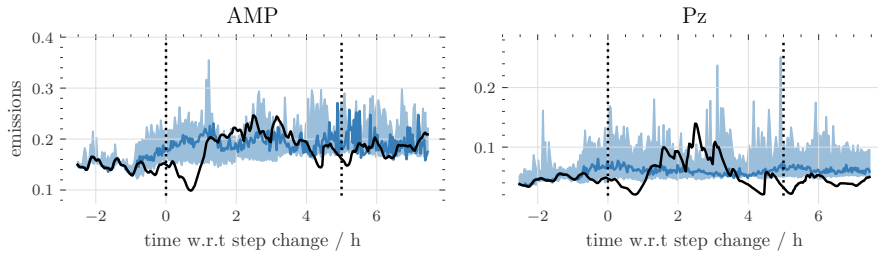


Figure 157: Causal impact analysis for day 3. Causal impact analysis for the step increase in flue gas temperature.

E.10.4 Day 4: Step decrease in lean solvent flow

The model shows relatively large prediction intervals and we cannot see a significant effect.

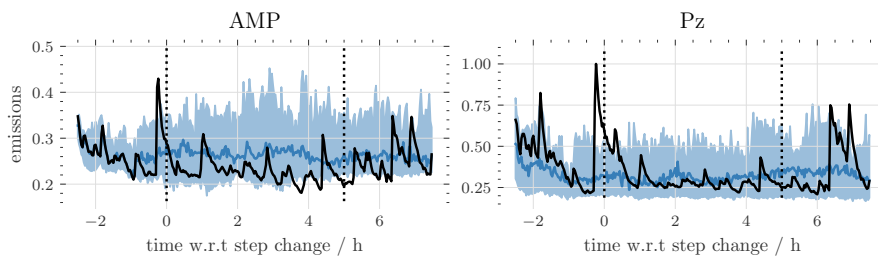


Figure 158: Causal impact analysis for day 4. Causal impact analysis for the step decrease in lean solvent flow.

E.10.5 Day 5: Lean solvent flow decrease and flue gas flow decrease

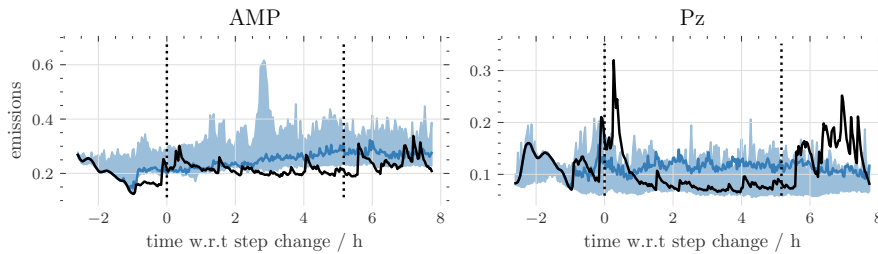


Figure 159: Causal impact analysis for day 5. Causal impact analysis for the lean solvent flow decrease and flue gas flow decrease.

E.10.6 Day 6: Step increase in lean solvent temperature

We can observe a slight decrease in Pz emissions compared to the baseline predictions.

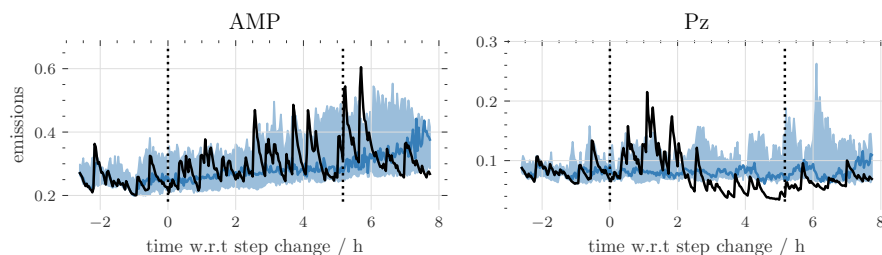


Figure 160: Causal impact analysis for day 6. Causal impact analysis for the step increase in lean solvent temperature.

E.10.7 Day 7: Lean solvent and water wash temperature increase

We observe a decrease in emissions for Pz and an increase in emissions for AMP compared to the baseline.

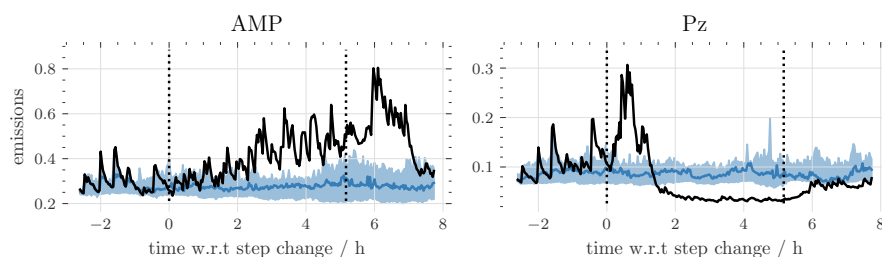


Figure 161: Causal impact analysis for day 7. Causal impact analysis for the lean solvent and water wash temperature increase.

E.11 EMISSION MITIGATION (SCENARIOS)

E.11.1 Method

ASSUMPTIONS Note that the modeling of scenarios is based on several assumptions:

- We assume that the dynamics of the systems are unchanged.
- We assume that other, possibly correlated, variables remain unchanged. (comparably to the assumptions in partial dependency plots).³³⁴ Note that we recompute composite variables, such as temperature differences, after performing the perturbations (before running the models).
- We assume that the cumulative change in emissions is a meaningful measure of the impact of the interventions.

Note that the maps shown in the main text are computed based on historical forecasts. That is, we use them as inputs for predicting the next timesteps the actually observed emissions together with the changed inputs in process parameters.

We need to impose such harsh assumptions since conventional black box explanation methods such as SHAP²⁶⁹ cannot directly be applied due to the time ordering of the inputs. Note that while this analysis does not provide a causal interpretation, it can still reveal important emission mechanism patterns.

ALGORITHM To compute the heatmaps, we create a mesh grid of values, typically of dimension 21×21 ranging from -20% to 20% relative change. For every point in the grid, we change the values of the two features on the axis according to the coordinate tuple (p_i, p_j) and run a forecast using our model, resulting in the time series $F(p_i, p_j, t)$. For the heatmap, we then compute

$$E(p_i, p_j) = \sum_t (F(p_i, p_j, t) - F(0, 0, t)) \quad (39)$$

For visualization purposes, we smooth the matrix $E(p_i, p_j)$ using a Gaussian filter. Note that the Gaussian filter is applied after centering the data according to eq. 39, which might lead to the center of the map not being centered at exactly zero. To remedy this, we recenter the maps after the smoothing step.

E.11.2 Additional maps

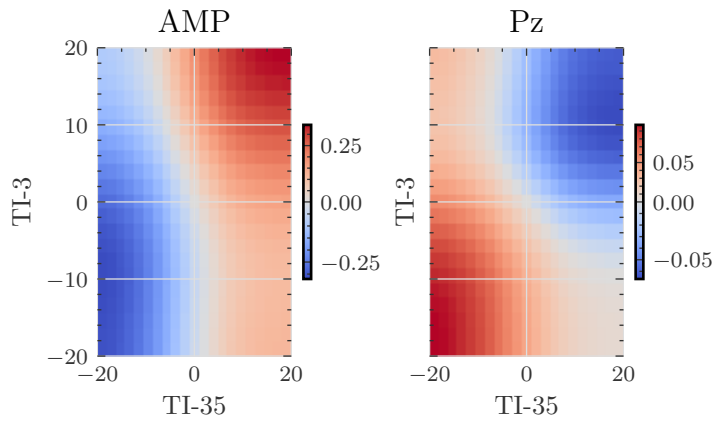


Figure 162: Scenario in TI-3 and TI-35. Emissions as a function of change (in percent) in TI-3 and TI-35 computed using historical forecasts of one-step-ahead predictions.

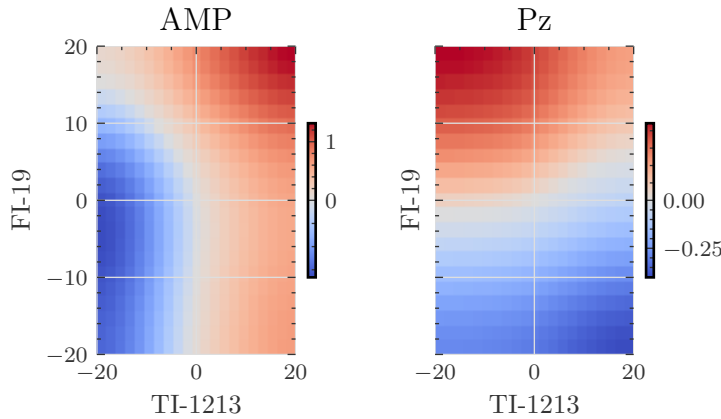


Figure 163: Scenario in FI-19 and TI-1213. Emissions as a function of change (in percent) in FI-19 and TI-1213 computed using historical forecasts of one-step-ahead predictions.

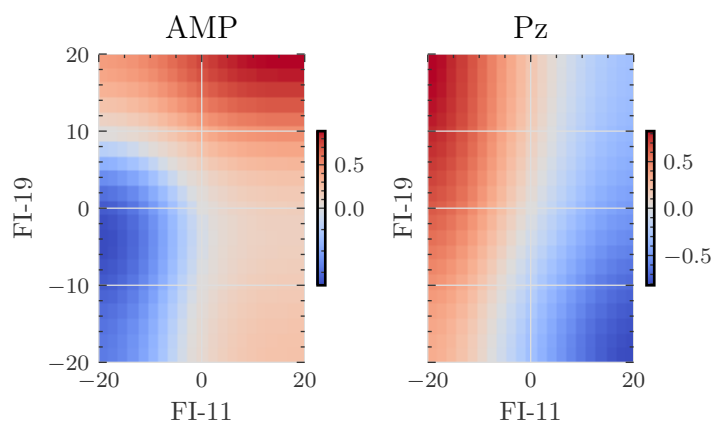


Figure 164: Scenario in FI-19 and FI-11. Emissions as a function of change (in percent) in FI-19 and FI-11 computed using historical forecasts of one-step-ahead predictions.

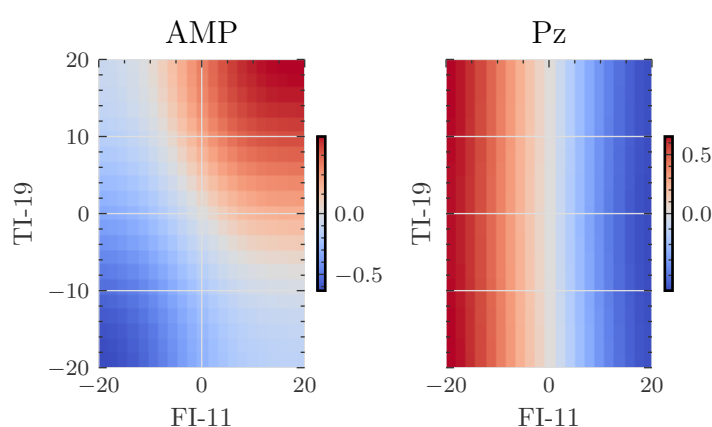


Figure 165: Scenario in TI-19 and FI-11. Emissions as a function of change (in percent) in TI-19 and FI-11 computed using historical forecasts of one-step-ahead predictions.

E.11.3 Using forecasts as input for new forecasts

The maps in the main text are computed using historical forecasts for one-step-ahead predictions. This implies that we assume that the emissions before a prediction are as measured—but with changed values for some covariates.

In this section, we lift this assumption and use the predicted historic emissions to compute forecasts.

The figures show that the overarching conclusion that the two amines behave differently for different possible mitigation measures still holds true under this perspective.

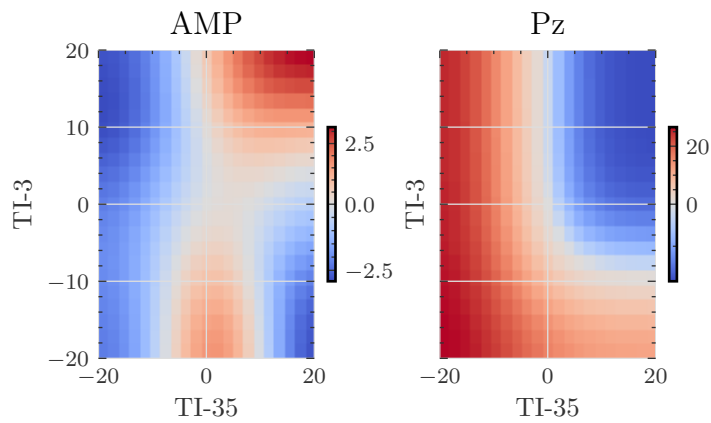


Figure 166: Scenario in TI-3 and TI-35. Emissions as a function of change (in percent) in TI-3 and TI-35 were computed using forecasted emissions as input for subsequent forecasts.

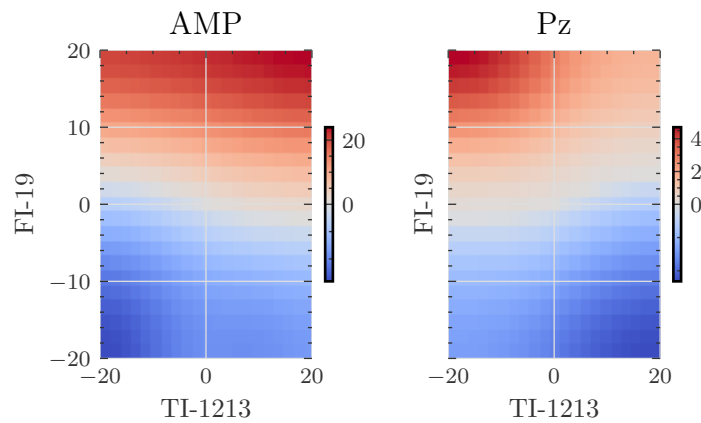


Figure 167: Scenario in FI-19 and TI-1212. Emissions as a function of change (in percent) in FI-19 and TI-1213 computed using forecasted emissions as input for subsequent forecasts.

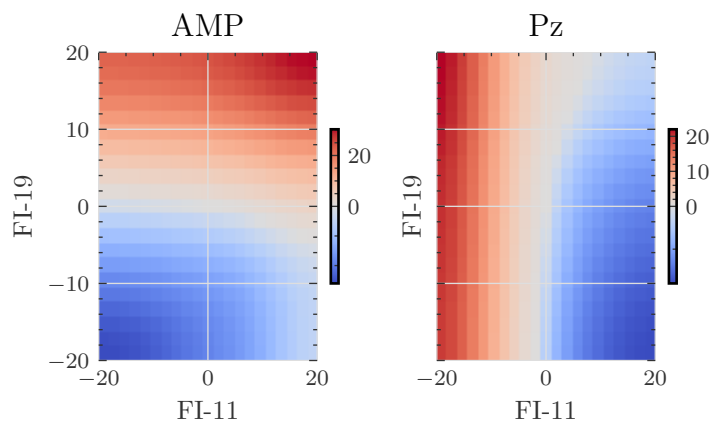


Figure 168: Scenario in FI-19 and FI-11. Emissions as a function of change (in percent) in FI-19 and FI-11 computed using forecasted emissions as input for subsequent forecasts.

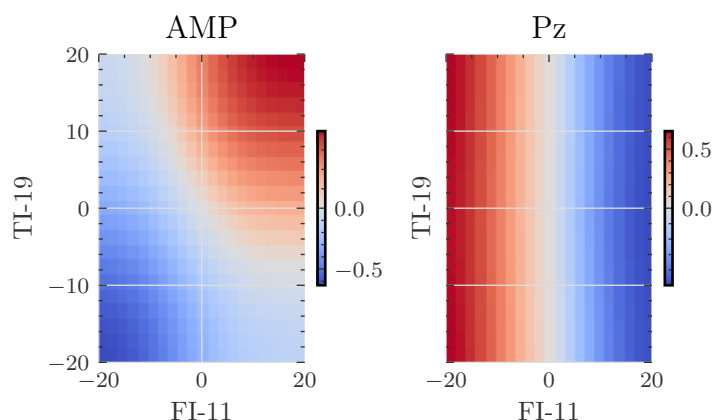


Figure 16g: Scenario in TI-19 and FI-11. Emissions as a function of change (in percent) in TI-19 and FI-11 computed using forecasted emissions as input for subsequent forecasts.

E.12 CAVEATS OF THE MODELING APPROACH

Clearly, there is no guarantee that our model learned causal relationships (i.e., true cointegration in contrast to spurious correlations), and the process parameter-emissions relationships shown in the heatmaps might be influenced by indirect influences. One could build graphical models (time series chain graphs)⁸²⁶ to counteract this.

For the interpretation of our results, this implies that any strong effect we predict, for example, in our heatmaps, might equally well be caused by a strongly correlated variable. We performed careful feature selection guided by discussion with domain experts to mitigate learning of spurious correlations. However, this also implies that the approach presented in this work is no silver bullet that can be applied to any dataset without data preprocessing.

E.12.1 Consistency of feature attributions

Our feature attribution techniques do not fulfill all requirements that Lundberg et al. enumerate.²⁶⁹ However, we observed that the trends, particularly the qualitatively different behavior of AMP and Pz, are consistent between models trained on different feature sets.

F

SUPPORTING INFORMATION
FOR “IS GPT-3 ALL YOU NEED
FOR LOW-DATA DISCOVERY IN
CHEMISTRY?”

F.1 METHODS

For all the results shown in the main text, we used the smallest ada variant of GPT-3 available via the OpenAI API. For fine-tuning, we used the same setting for all case studies (eight epochs, learning rate multiplier of 0.02).

Error bands show, if not otherwise indicated, the standard error of the mean.

F.1.1 Data efficiency comparison

To compare the data efficiency of the GPT-3 models with our baselines, we fitted all learning curves to powerlaws ($-a \exp(-bx) + c$). We then used those powerlaws to find the point the best-performing baseline shows the same performance as the best GPT-3-based approach at the first learning curve point (that performs better than random, as measured using the κ metric).

F.1.2 Validity checks

To check the validity of the generated SMILES, we use the `is_valid` method from the GuacaMol package,¹³⁹ which effectively considers a SMILES as valid if it can be parsed using RDKit.

F.2 APPLICATIONS AND DATASETS

In this work, we tested the GPT-3 model for different chemistry and material science applications. We selected those applications for which successful machine-learning approaches have been developed. This allows us to assess the performance of our GPT-3 approach with the state-of-the-art. We have divided these applications into three categories: properties of molecules, properties of materials, and reactions.

F.2.1 Molecules

PHOTOSWITCHES Photoswitches are molecules that reversibly change their structure—and thus polarity—upon irradiation with light. This behavior can be useful for various applications. Incorporated into drug molecules, for instance, photoswitches might be used to control the activity of the drug molecule. Besides that, they might also be incorporated into molecular machines or used for molecular energy and information storage. See Crespi et al.⁸²⁷ for a review of azobenzene photoswitches.

One would like to tailor the adsorption of a photoswitch molecule for a given application. For example, for therapeutic applications, one would like to red-shift the adsorption band to use lower energy light and minimize the potential for radiation damage. Often, one also wants to achieve some separation between the adsorption of the *E* and *Z* isomers, to ensure that they can be triggered independently.

One can tune the transition wavelength by modifying the structure. We have a reasonable intuition if we make small modifications to the structure. However, as soon as we make multiple modifications, this intuition is of limited use Griffiths et al.¹⁴³. Hence, one has to rely on computational chemistry, via TDDFT, to obtain some guidance. However, these quantum calculations require expert knowledge and are computationally prohibitive for large-scale screenings. This motivated Griffiths et al. to develop a machine-learning approach to predict the transition wavelengths Griffiths et al.¹⁴³

QMUGS A relevant property for many (opto)electronic application is the energetic difference between the HOMO and LUMO. Isert et al.⁴¹⁴ computed this (and related properties) for multiple conformers of about 500,000 molecules using different levels of theory.

SOLUBILITY Aqueous solubility is an essential characteristic of molecules, particularly for pharmacological applications. If a molecule does not dissolve in water, it will have poor bioavailability. Hence, aqueous solubility is an important factor in evaluating potential drug molecules.⁸²⁸

An experiment by Pat Walters⁴¹⁰ inspires our solubility case study. In this experiment, Walters compared different models trained on the ESOL dataset of⁴⁰⁸ and tested them on a dataset of pharmacologically relevant molecules, the DLS-100 dataset.⁴⁰⁹ The models under consideration were a refitted version of the original ESOL, RFs, a graph neural networks (GNNs) Weave modules,⁸²⁹ as well as a single-layer graph convolutional neural networks.⁸³⁰

HYDRATION FREE ENERGIES A property closely related to solubility is the hydration free energy. It describes the change in free energy when a molecule is transferred from the gas phase into water. On a practical level, they are very useful for benchmarking force fields. However, they can also give insights into solvation mechanisms.⁴⁰⁷

LIPOPHILICITY Similar to solubility, lipophilicity is an essential factor in pharmacokinetics.⁸³¹ It is typically described using the water-octanol partition coefficient (logD). Therefore, many works have focussed on predicting this coefficient.

ORGANIC PHOTOVOLTAICS Organic photovoltaics (OPV) to provide photovoltaic systems that can be produced from Earth-abundant elements using low-energy techniques. A widely investigated device architecture comprises a p-type molecule or polymer and an n-type fullerene, forming a bulk heterojunction framework. A key performance indicator for OPV is the power conversion efficiency (PCE), i.e., the ratio of output power to input power.

F.2.2 Materials

METAL-ORGANIC FRAMEWORKS MOFs are one of the most exciting classes of materials as they promise to provide a framework for systematically designing materials across different scales.^{16,94,832} They are composed of metal clusters connected via strong bonds to organic linkers. Due to their tunable porosity and the

potential to tune the chemistry to a given application, they have been explored for a wide array of applications ranging from gas storage and separation, over sensing, to catalysis.

Henry coefficients For gas separations, the Henry coefficient is a key performance indicator. For diluted streams, the Henry coefficient multiplied by the pressure gives the number of adsorbed molecules. In various studies, it has been shown to be a valuable indicator for gas separation applications, e.g., carbon capture.¹⁴⁹

Heat capacities If solid sorbents such as MOFs are used in a carbon capture process, they must be heated to regenerate. The amount of needed heat crucially depends on the heat capacity of the materials. Moosavi et al.¹²⁵ has shown that neglecting this factor can change screening results if process performance metrics are considered.

Water stability If a MOFs is used in an industrial process, it will invariably be exposed to some form of water: For example, from a humid flue gas stream or the steam used to regenerate the material. A useful MOFs must remain stable under such conditions. Unfortunately, as Burtch et al.⁸³³ reviewed, water stability is a complex phenomenon in which a material might be kinetically or thermodynamically stable. Since we do not understand the degradation mechanism and the complex chemistry of MOFs with large unit cells, this question can currently not be (routinely) addressed using tools from computational chemistry.

POLYMERS As another material case study, we considered linear polymers in a coarse-grained representation.²⁰⁷ Those polymers are representative of dispersants that have various applications in industry, for instance, to stabilize the color brightness of pigments. An important performance parameter for this application is the free energy of adsorption of the polymer onto a model surface (e.g., of a pigment particle). Jablonka et al.²⁰⁷ computed such free energies using mesoscale simulations. In order to build a surrogate model for the expensive simulations, Jablonka et al.²⁰⁷ developed a ML approach based on features extracted from the monomer sequence. Besides the composition, this feature set also included cluster size statistics and measures of the entropy of the sequence and was manually tuned for optimum performance on this task.

HIGH-ENTROPY ALLOYS High entropy alloys (HEAs)^{834–836} are materials that are composed of multiple elements (often > 5) in roughly equal proportions. Since their discovery, they have attracted much interest due to their promising properties, with some materials overcoming the strength-ductility tradeoff. In particular, Yeh et al.⁸³⁴ proposed that the configurational entropy might stabilize the solid solutions at the expense of intermetallics (which often tend to be brittle, note, however, that the configurational entropy rule does not fully stand the test).⁸³⁶ However, they also span a large chemical space that is difficult to explore using conventional techniques. A critical question is what phase(s) will form for a given alloy composition. Before the discovery for HEA Hume-Rothery put forward a set of general rules for the potential for forming solid solutions of binary alloys. Since then, various approaches have been put forward to predict the phase formation of HEAs, including a ML approach by Pei et al.⁴⁰⁰, who constructed feature vectors based on atomic properties. Here, we reuse their dataset, but simply provide GPT-3 with the composition, e.g., $\text{Ag}_{0.05}\text{Zr}_{0.95}$ as input.

MATBENCH TASKS To ensure a fair comparison to top-performing models in material science, we also considered the tasks from the MatBench suite that are based on compositions.²¹²

Bulk metallic glass formation ability The ability of a material to form glasses is known as the glass formation ability and is linked to the chemical composition. Dunn et al.²¹² extracted a dataset of composition and the bulk metallic glass formation ability from the Landolt-Börnstein collection.⁴¹⁶

Metallicity In addition, we also considered classifying compositions as metallic or non-metallic. For this Dunn et al.²¹² extracted a dataset reported by Zhuo et al.⁴¹⁷

Experimental band gaps Zhuo et al.⁴¹⁷ also extracted a dataset of compositions and band gaps of inorganic solids from Zhuo et al.⁴¹⁷

Yield strength of steel For structural materials, it is essential to know at which stress it ceases to show elastic behavior. That is, above the yield point, a deformation will be permanent. Dunn et al.²¹² extracted this information for steels from Citrine informatics.

F.2.3 Reactions

Very central to the central science are chemical reactions. A lot of time is often spent optimizing conditions to increase the yield. Recently, there have been many attempts to expedite this process using ML approaches such as Bayesian optimization (BO). In many works, it was found that one-hot encoding performs equivalent to chemically informed representations.^{31,837,838} However, it would be much more convenient if the reactants were simply provided as text without preprocessing. Here, we focus on two datasets. First, experimentally determined yields of Pd-catalysed Buchwald-Hartwig C-N crosscouplings reported by Ahneman et al.⁴¹⁸ and, second, yields for Pd-catalysed Suzuki-Miyaura C-C cross-couplings reported by Perera et al.⁴¹⁹

F.3 FINE-TUNING PARAMETERS

As an initial experiment, we investigated the impact of a range of fine-tuning parameters. We focussed on balanced binary classification on the photoswitch using SMILES.

In Figure 170 we observe that the fine-tuning parameters can have a pronounced effect on the classification performance. Particularly for only a few passes through the data (low number of epochs) and a low learning rate (η) the resulting model might even perform worse than random guessing.

Interestingly, we do not find the largest model to perform consistently better.

We decided to use the same settings for all subsequent experiments instead of optimizing them for every experiment. While this might limit the performance we observe, it also avoids overfitting.

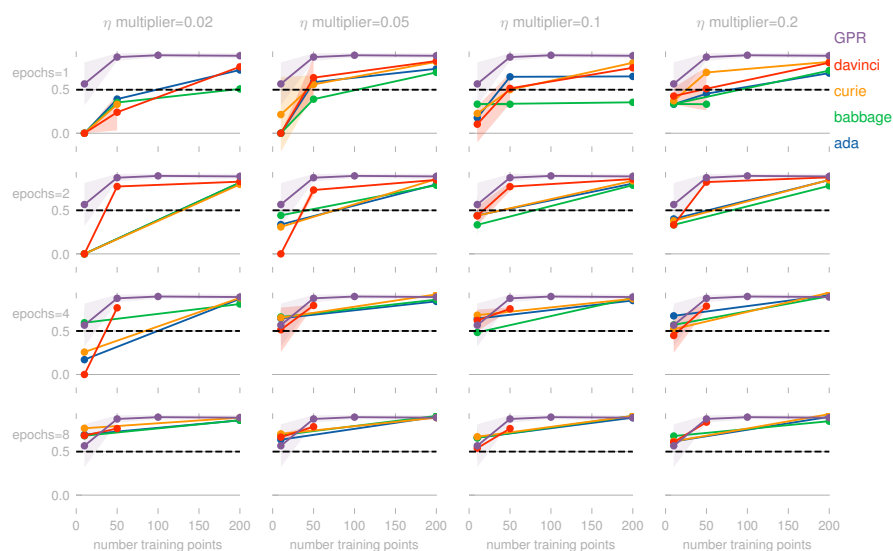


Figure 170: Influence of fine-tuning parameters on the model performance.

Note that the model must also use the few examples we provide to learn the prompt structure. From Dinh et al.³⁹⁸ we know that pre-training with prompts filled with random data can help the model learn.

F.4 INFLUENCE OF THE MOLECULAR REPRESENTATION

There is not one unique way of representing molecules in text form. While the IUPAC name might have widespread use for the communication of synthesis protocols or papers, cheminformatics focussed on line representations such as SMILES,⁴²³ INChI, and recently, SEFLIES.⁴²⁴ For an overview, see Krenn et al.⁴²⁵

To understand possible representation effects better, we investigated other possible representations of the molecules in the photoswitch dataset. First, we fragmented the molecules using extended functional groups (EFGs).⁸³⁹ We then directly used those fragment SMILES as representation but also investigated the removal of chemical information using a numerical encoding. In addition, we also removed explicit chemical information from SEFLIES by replacing the characters using a numerical encoding. To investigate potential sequence length effects, we also investigated padding the SEFLIES characters. In Figure 171 we show that both chemistry and the sequence length influence the predictive performance. Removing explicit chemistry information tends to decrease the performance, and reshaping it into relevant “buckets” via fragments tends to increase the performance. This indicates that while the fine-tuning of GPT-3 gives performance that is competitive with strong baselines, unlocking the ultimate power still requires, as is the case with all models, fine-tuning of the representation. In the case of LLMs, this implies tuning prompt and string representations in contrast to conventional feature engineering.

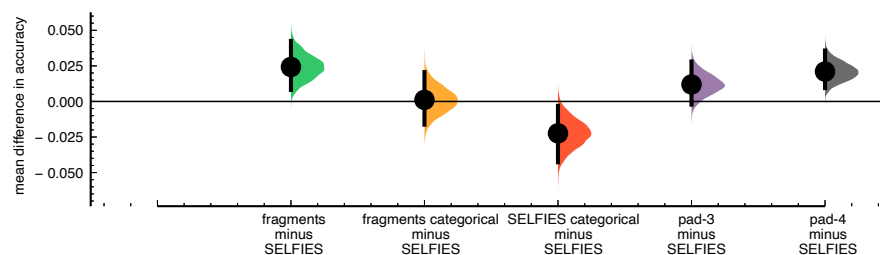


Figure 171: Mean effect sizes for accuracy on the photoswitch case study for different representations. Top row shows the accuracy obtained in independent runs with different representations. The bottom row shows bootstrapped mean effect sizes versus the SELFIES baseline. We created the figures using the DABEST library.⁸⁴⁰

F.5 FEW SHOT REASONING

As Brown et al.³⁹³ showed, LLMs are few-shot learners that can generalize to new tasks without any gradient-based learning (e.g., fine-tuning).

For this reason, we also investigated this setting. To do so, we focussed on classification on the photoswitch dataset and prompted different versions of GPT-3 with prompts of the form

I am a highly intelligent question answering bot that answers questions about transition wavelengths of photoswitch molecules.

Q: CC(C=C(N(CCC#N)CCO)C=C1)=C1/N=N/C2=CC=C(C(F)(F)F)C=C2

A: 421.0

Q: OC1=C([N+])([O-])=O)C=C([N+])([O-])=O)C=C1/N=N/C2=C(O)C=CC(C)=C2

A: 400.0

Q: O=[N+])([O-])C1=CC=C(/N=N/C2=CC=C(NCCC#N)C=C2)C=C1

A: 455.0

Q: FC1=CC=C(/N=N/C2=CC=CC=C2)C=C1

A: 322.0

Q: CCN(CC)C(C=C%21)=CC=C%21/N=N/C%22=CC=C(N%23CCOCC%23)C([H])=C%22

A: 417.0

Q: CN(C=N1)C=C1/N=N/C2=CC=CC=C2

A: 336.0

Q: CCN(CC)C(C=C1)=CC=C1/N=N/C2=C(C#N)C=C(C#N)C=C2

A: 515.0

Q: CN(C)C(C=C1)=CC=C1/N=N/C2=CC=CC=C2[N+])([O-])=O

A: 440.0

Q: CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(O)=O)C=C2

A: 342.0

Q: CC(C=C(N(CCC#N)CCO)C=C1)=C1/N=N/C2=CC=C(C(C)=O)C=C2

A: 412.0

Q: CC(C=C(N(CCC#N)CCO)C=C1)=C1/N=N/C2=C(F)C=CC=C2

As evident from Figure 172 and Figure 173 we find that also with this approach, we find good results, competitive, or even outperforming the GPR baseline (also see Appendix F.6). Interestingly, we also observe a stronger dependence on the representation of the molecules when we use this prompt-based approach without fine-tuning. However, a fundamental limitation of this approach is that the number of examples one can provide is limited by the context length of the model. For this reason, the learning curves also add at different points (with larger models having larger context sizes and different representations needing a different number of tokens).

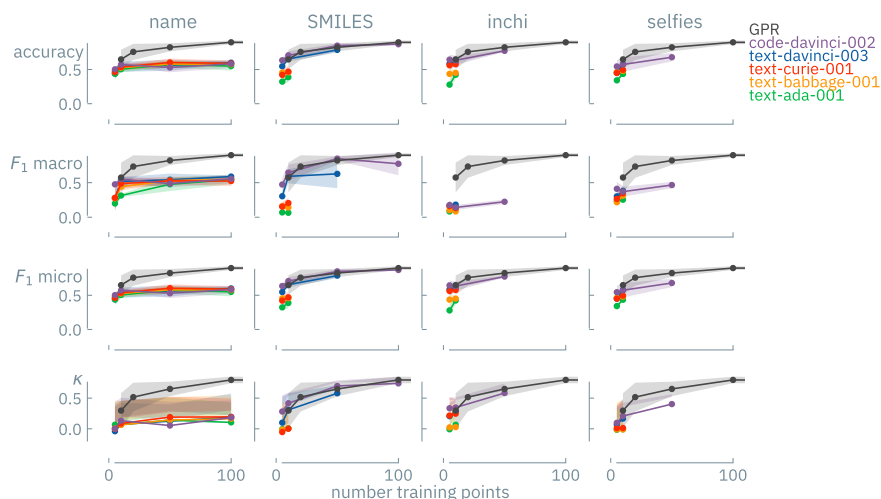


Figure 172: Classification metrics on the photoswitch dataset for binary classification with few-shot prompts. Columns indicate different representations, colors different models.

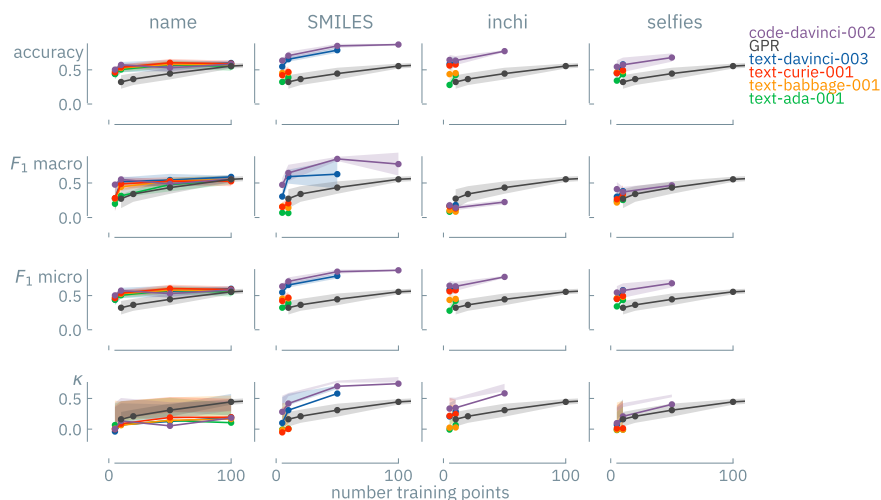


Figure 173: Classification metrics on the photoswitch dataset for 5-class classification with few-shot prompts. Columns indicate different representations, colors different models.

These encouraging results motivate further work using other prompting strategies, including soft prompting,⁸⁴¹ such as instruction prompt tuning.⁸⁴²

F.6 CLASSIFICATION EXPERIMENTS

F.6.1 Note about baselines

For the classification case studies, we use MolCLR and tabular Prior-Data Fitted Network (PFN) (TabPFN) as baselines in most cases.

MOLCLR Wang et al.⁴⁰² proposed the Molecular Contrastive Learning of Representations via Graph Neural Networks approach in which GNNs are pre-trained on large unlabeled datasets using a contrastive loss and graph augmentations such as atom masking, bond deletion, and subgraph removal. They could show that this pre-training improves the performance of GNNs on various property prediction tasks. In this work, we fine-tune MolCLR on all tasks with the default settings reported by the original authors.

TABPFN TabPFN⁴⁰⁴ is a transformer (25.82 million parameters) that has been pre-trained on millions of synthetic datasets (D_i sampled from some prior $p(\mathbf{D})$ with the task of predicting held-out points with a forward pass (i.e., it predicts posterior predictive distribution (PPD) $q_\theta(y_{\text{test}}|x_{\text{test}}, D)$). This can be thought of as gradient-based meta-learning.

GPR WITH FRAGPRINTS For datasets that have SMILES available, we also use GPRs models with Tanimoto kernels with “fragprint” descriptors, which are concatenations of fragment features and fingerprints.^{143,843}

For specific case studies, there are additional baselines we took from the literature and which we describe in the corresponding section.

F.6.2 Photoswitches

The learning curves for classification are shown in Figure 174 and Figure 176. Particularly for the binary classification, the GPT-3 model fine-tuned on IUPAC-names outperforms the baselines.

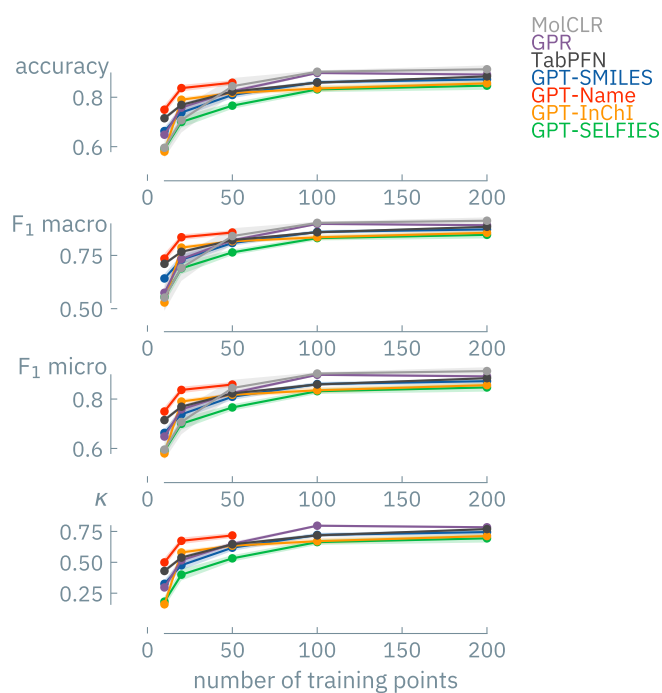


Figure 174: Learning curves for binary classification on the photoswitch dataset. We did a balanced split in two classes to predict the π - π^* transition wavelength of the *E* isomers.

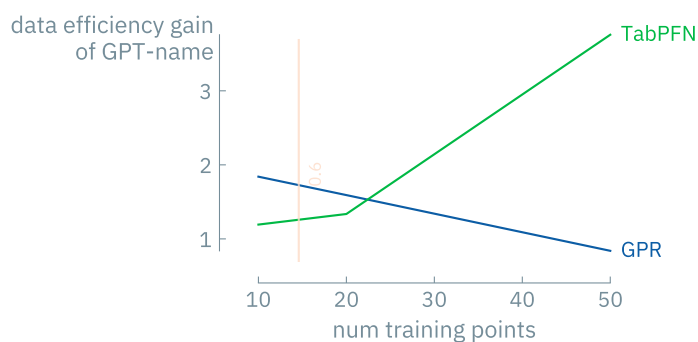


Figure 175: Learning curve intersection points for binary classification on the photoswitch dataset. Vertical lines indicate the κ scores of the GPT-3 model.

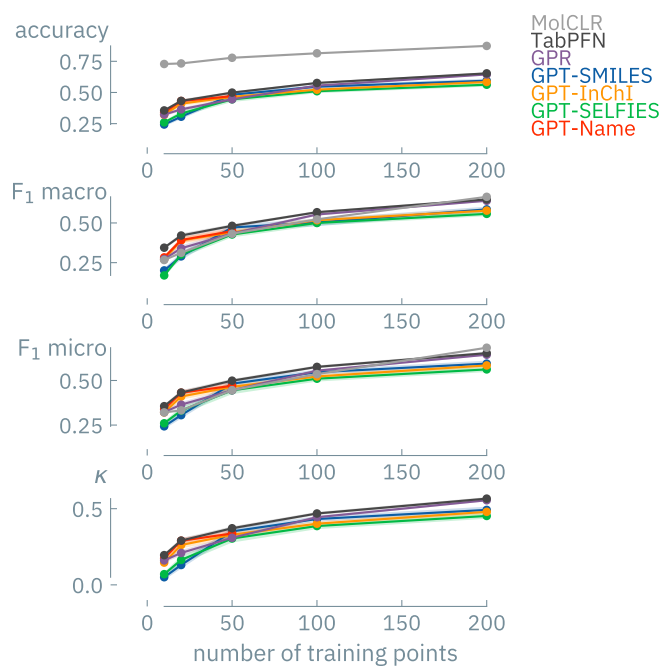


Figure 176: Learning curves for 5-class classification on the photoswitch dataset. We did a balanced split in five classes to predict the π - π^* transition wavelength of the *E* isomers.

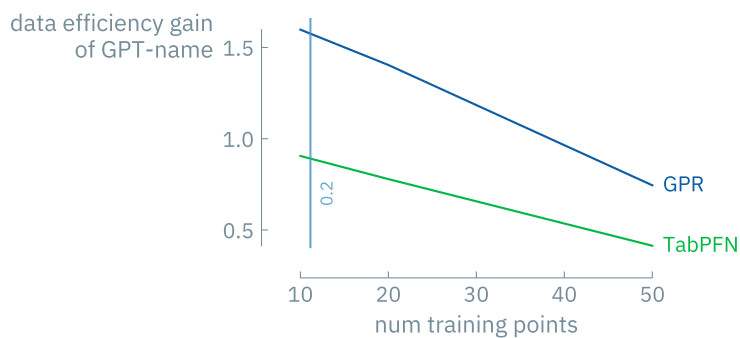


Figure 177: Learning curve intersection points for binary classification on the photoswitch dataset. Vertical lines indicate the κ scores of the GPT-3 model.

F.6.3 Free energy of solvation

The learning curves for the classification tasks are shown in Figure 178 and Figure 180. Here, we find GPT-3 fine-tuned using the SEFLIES representation to perform well.

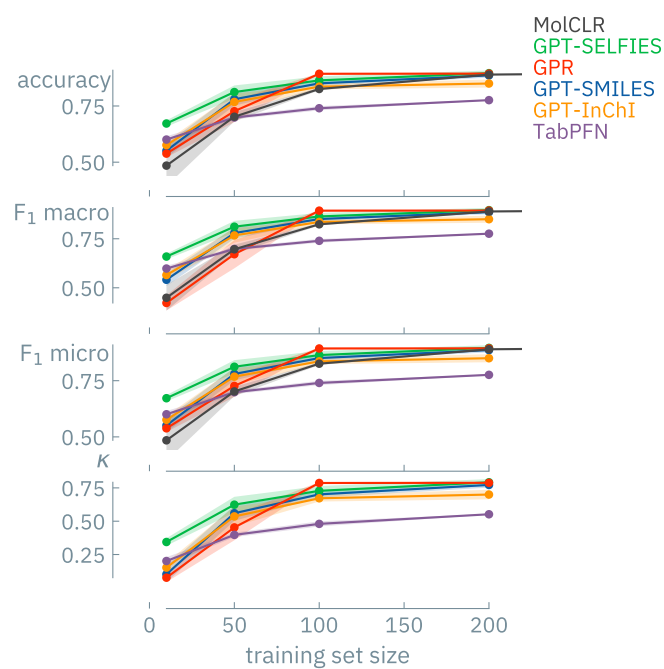


Figure 178: Learning curves for binary classification on the FreeSolv dataset. We performed a balanced split into two classes.

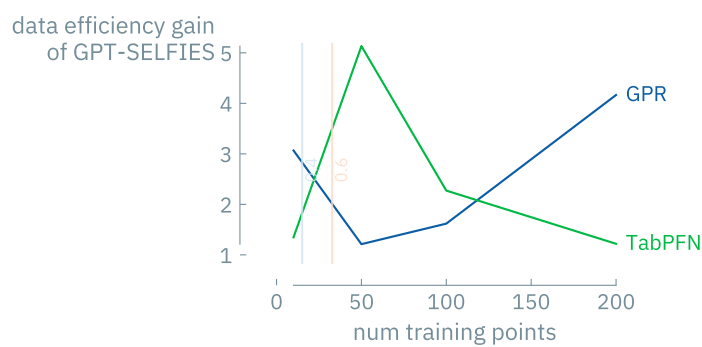


Figure 179: Learning curve intersection points for binary classification on the FreeSolv dataset. Vertical lines indicate the κ scores of the GPT-3 model.

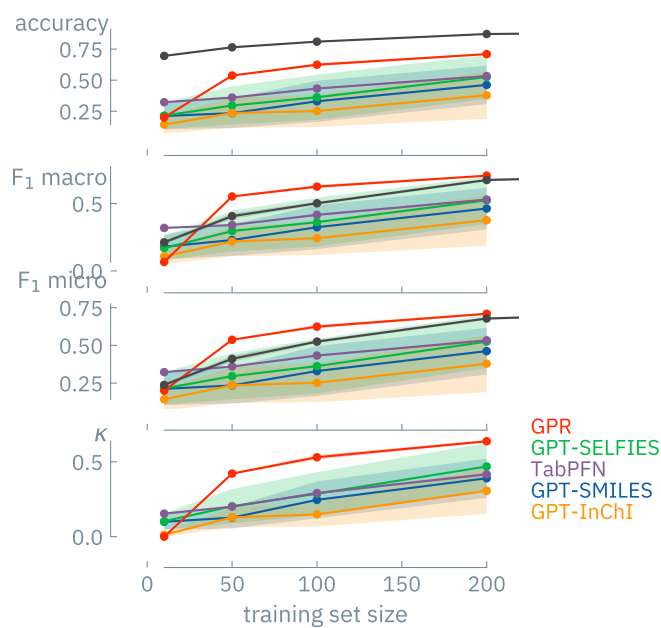


Figure 180: Learning curves for 5-class classification on the FreeSolv dataset. We performed a balanced split into five classes.

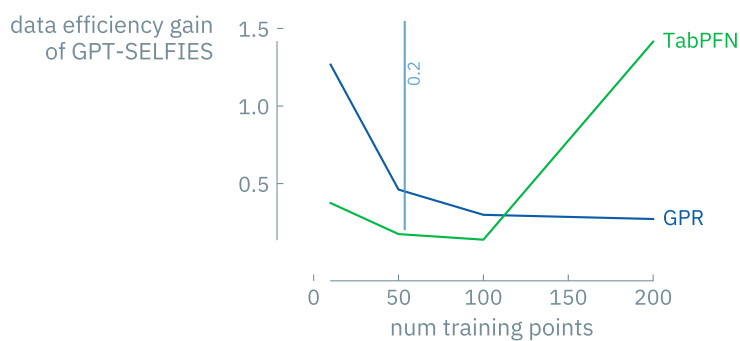


Figure 181: Learning curve intersection points for 5-class classification on the FreeSolv dataset. Vertical lines indicate the κ scores of the GPT-3 model.

F.6.4 Lipophilicity

The learning curves for the classification tasks are shown in Figure 182 and Figure 184.

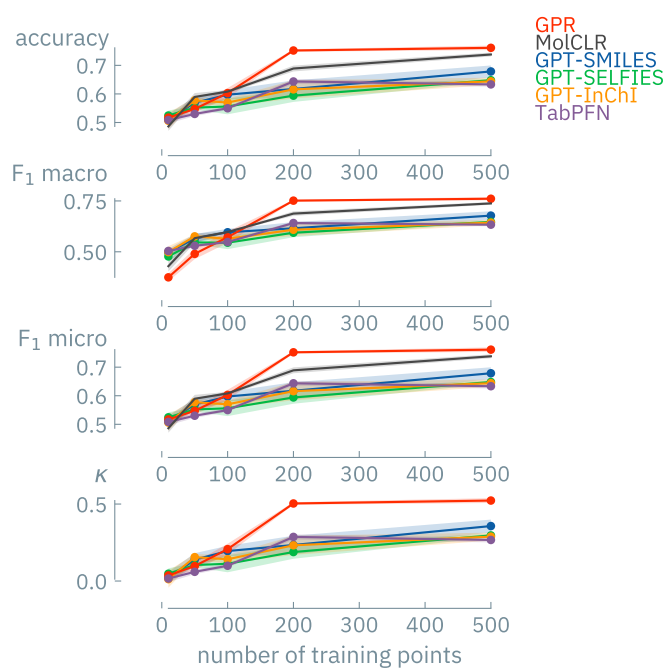


Figure 182: Learning curves for binary classification on the lipophilicity dataset.

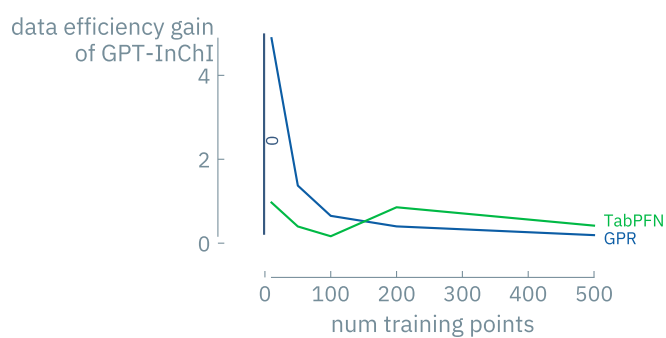


Figure 183: Learning curve intersection points for binary classification on the lipophilicity dataset. Vertical lines indicate the κ scores of the GPT-3 model.

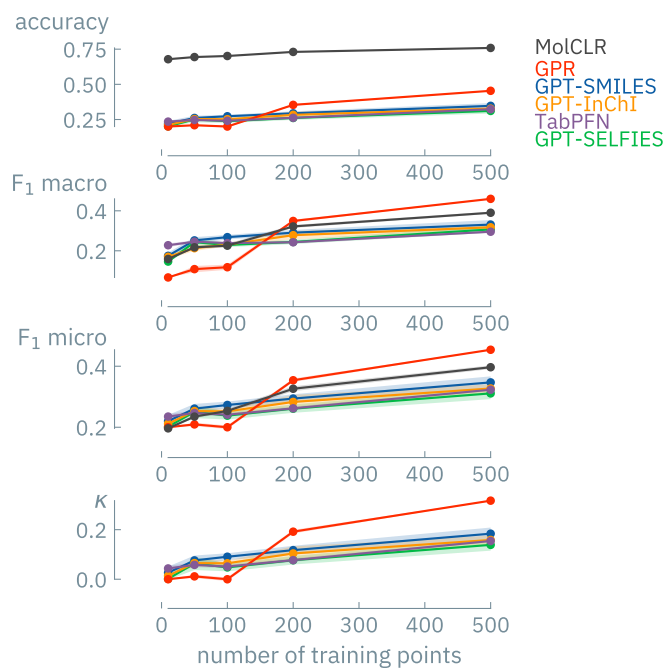


Figure 184: Learning curves for 5-class classification on the lipophilicity dataset.

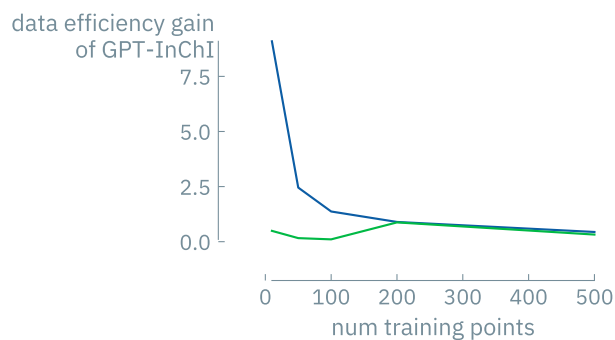


Figure 185: Learning curve intersection points for 5-class classification on the FreeSolv dataset. Vertical lines indicate the κ scores of the GPT-3 model.

F.6.5 HOMO-LUMO gaps

First, it is essential to note that different conformers have different gaps between HOMO and LUMO. The distribution of the standard deviation of the HOMO-LUMO gaps for different conformers of a molecule in the QMUGs dataset is shown in Figure 186.

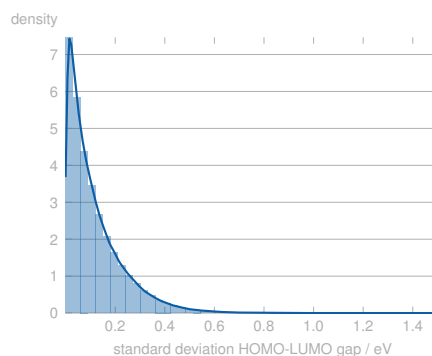


Figure 186: Distribution of standard deviations of the HOMO-LUMO gap of the three conformers of the molecules in the QMUGs dataset.

Since we base our model on line representations, we cannot resolve those effects and train our models on the average HOMO-LUMO gap of the three conformers reported by Isert et al.⁴¹⁴ The learning curves for the classification tasks are shown in Figure 187 and Figure 189.

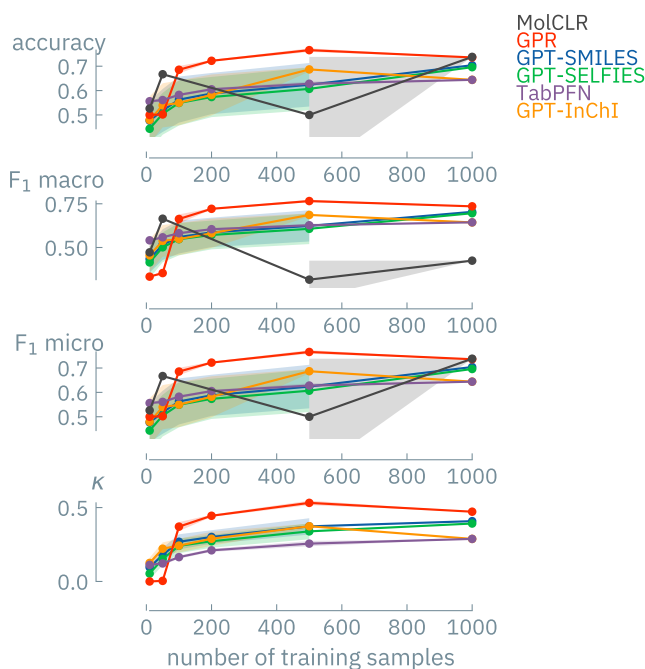


Figure 187: Learning curves for binary classification of HOMO-LUMO gaps on the QMUG dataset.

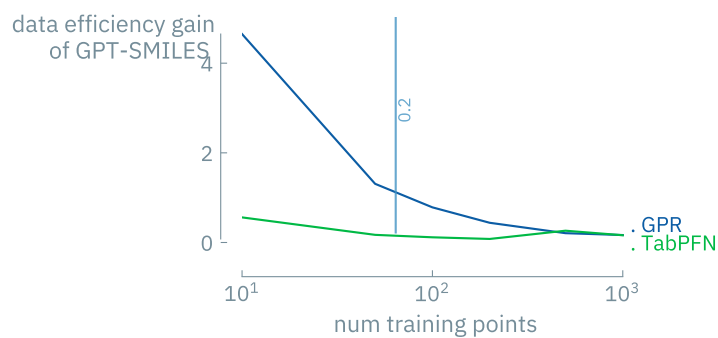


Figure 188: Learning curve intersection points for binary classification on the QMUG dataset.

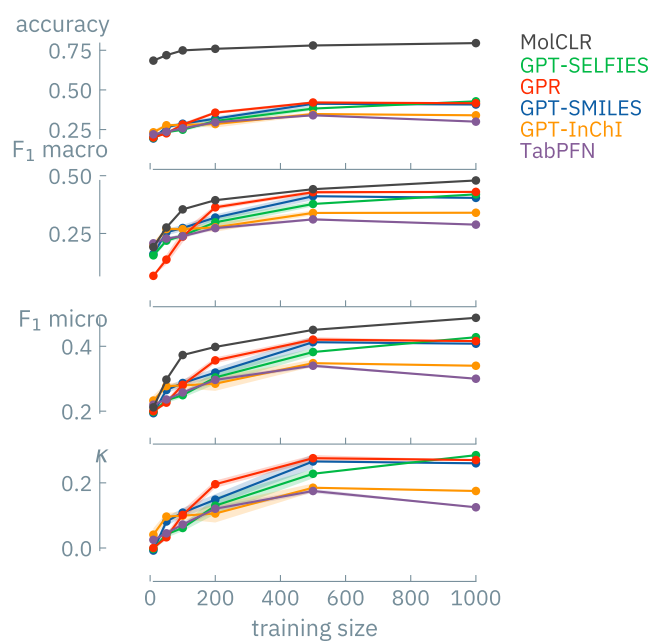


Figure 189: Learning curves for 5-class classification of HOMO-LUMO gaps on the QMUG dataset.

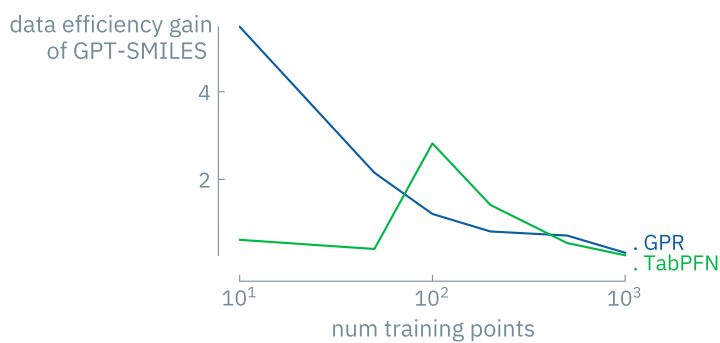


Figure 190: Learning curve intersection points for 5-class classification on the QMUG dataset.

F.6.6 Photoconversion efficiencies in organic photovoltaics

Nagasawa et al.⁴¹⁵ reported a dataset of performance of around 1000 conjugated molecules compiled from the experimental literature. They also reported machine learning models for which they found good predictive performance for RF models using an extended connectivity fingerprint (ECFP). In their dataset, there are some duplicated entries which we grouped and aggregated using the mean value. Since the authors did not report the hyperparameters they used, we also implemented 5-fold cross-validated hyperparameter optimization on a RF model as a baseline and completed with the baselines we use throughout all case studies.

The learning curves for the classification tasks are shown in Figure 191 and Figure 192.

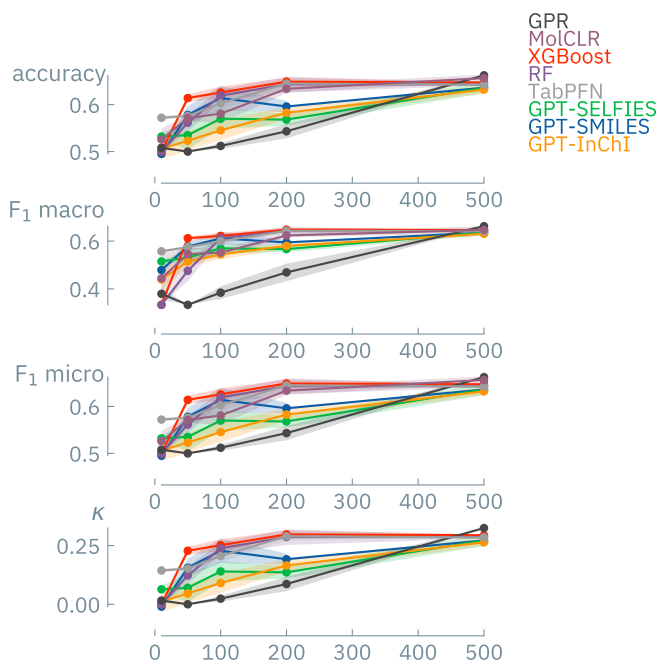


Figure 191: Learning curves for binary classification on the OPV dataset. We performed a balanced split into two classes based on the average PCE.

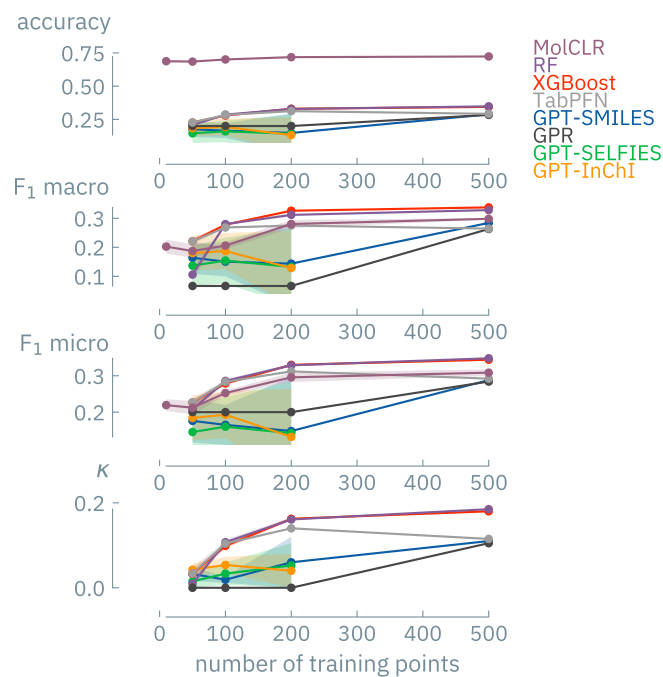


Figure 192: Learning curves for 5-class classification on the OPV dataset. We performed a balanced split into five classes based on the average PCE.

F.6.7 Solubility

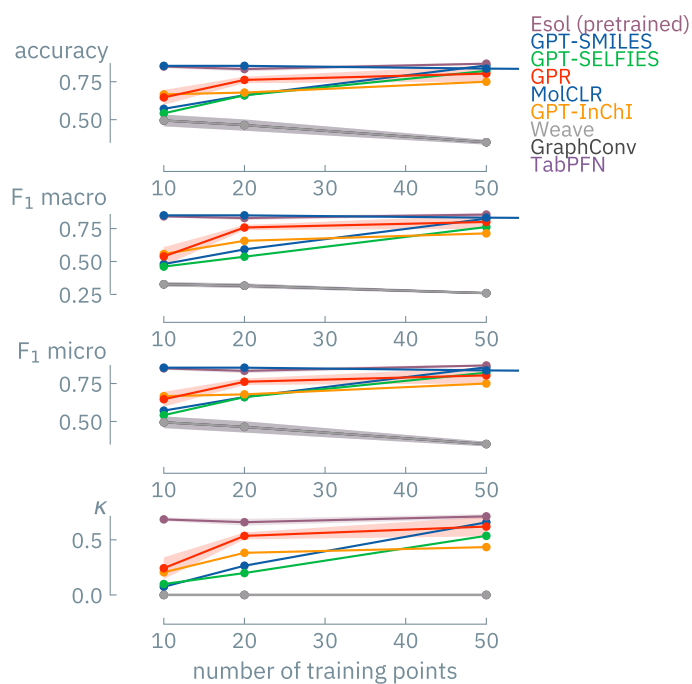


Figure 193: Learning curves for binary classification on the solubility dataset.

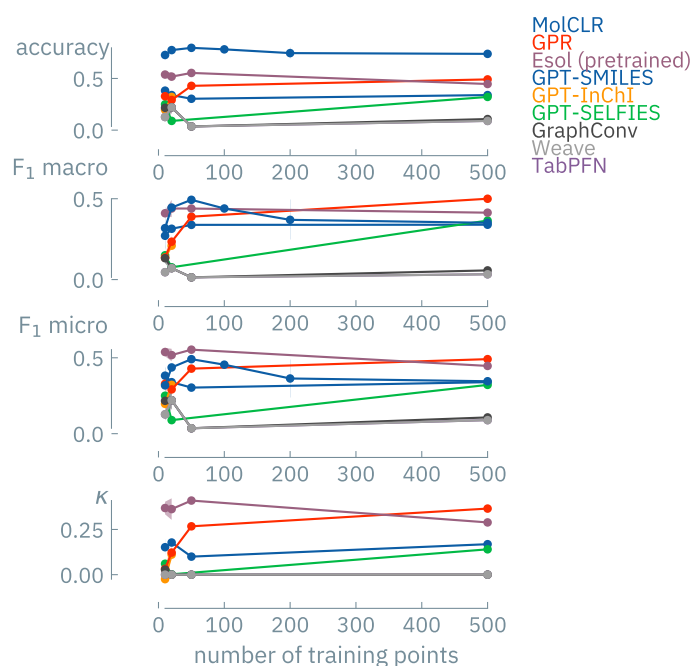


Figure 194: Learning curves for five-class classification on the solubility dataset.

f.6.8 MOF Henry coefficients

Carbon dioxide and methane

To fairly compare our results to a strong baseline, we reuse the features and models reported by Moosavi et al.¹³² Note that the models reported by Moosavi et al.¹³² were extensively tuned on the target task with specifically engineered features. Since the current implementation of TabPFN can only handle 100 features, we selected the 100 most relevant features according to the feature importance of a RF model.

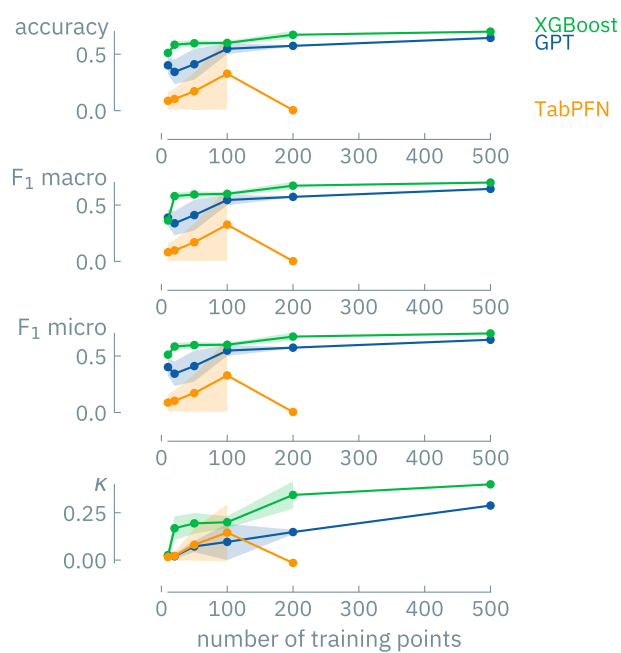


Figure 195: Learning curves for binary classification of CO_2 Henry coefficients.

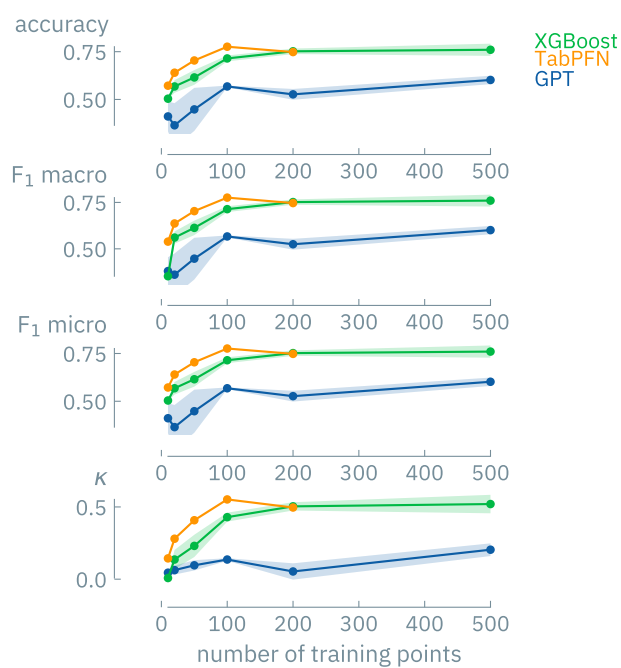


Figure 196: Learning curves for binary classification of CH_4 Henry coefficients.

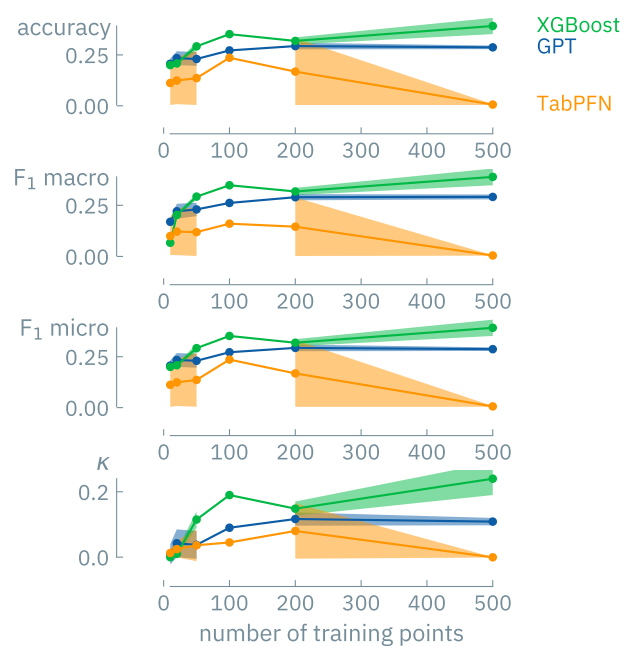


Figure 197: Learning curves for 5-class classification of CO_2 Henry coefficients.

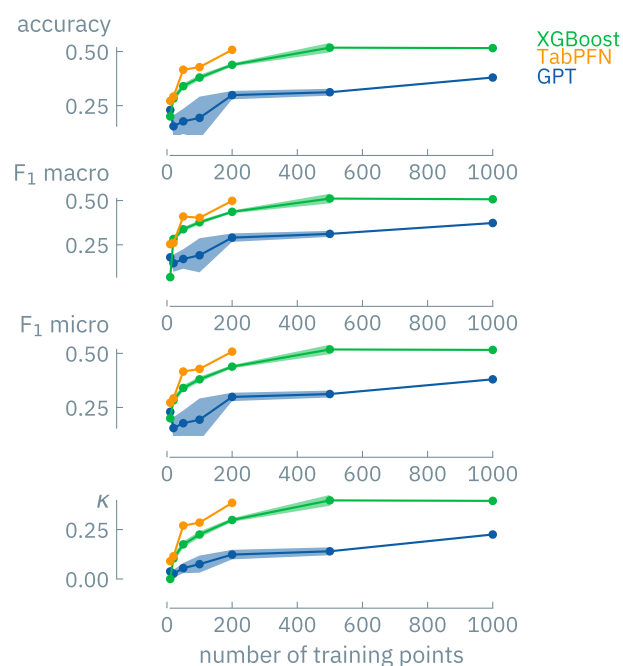


Figure 198: Learning curves for 5-class classification of CH_4 Henry coefficients.

r.6.9 MOF heat capacities

We use data reported by Moosavi et al.¹²⁵. We also use the model reported by Moosavi et al.¹²⁵, for which they extensively optimized hyperparameters. Notably, this model can achieve good data efficiency because it has been trained on local environments. Additionally, this model is a bootstrapped ensemble, which we replicated for our baseline. To ensure a fair comparison, we only trained on MOFs and

not COFs and zeolites. Additionally, we only considered those MOFs for which we could determine a mofid.¹⁵² Note that, similar to Moosavi et al.¹²⁵ we did not utilize advanced splitting strategies other than stratification on the gravimetric heat capacity. Additionally, we performed the split on the structures and not on the sites and dropped duplicates based on the mofid.

Additionally, we investigated a composition-only approach (reduced formula of the primitive cell). As a baseline for this, we considered CrabNet.⁴⁰¹

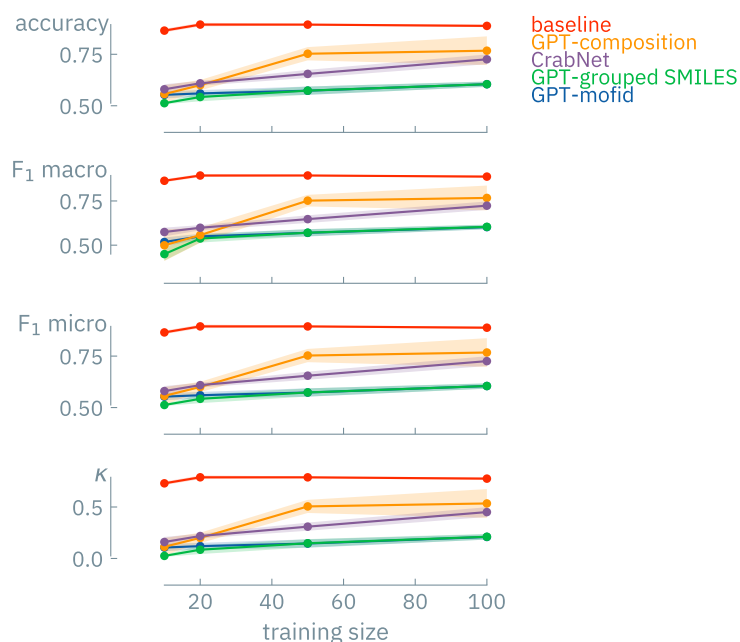


Figure 199: Learning curve for binary classification of MOF heat capacities.

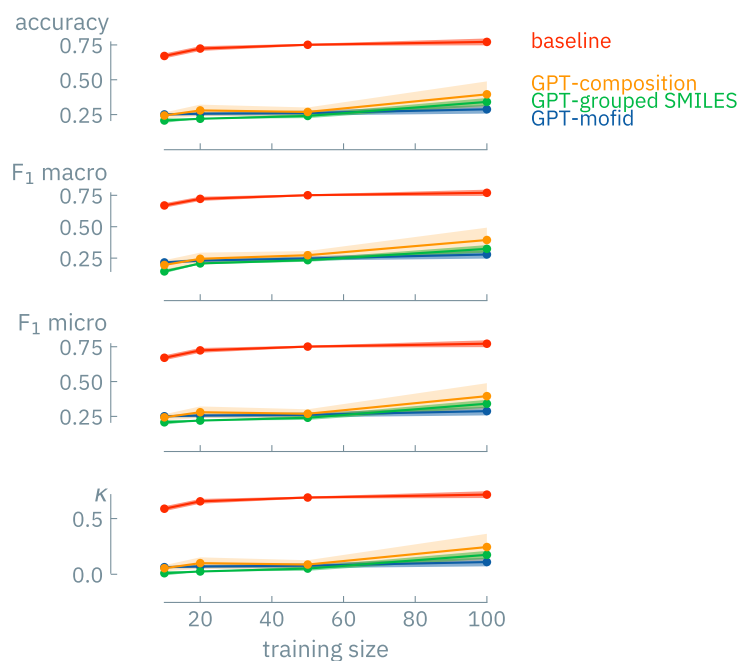


Figure 200: Learning curve for 5-class classification of MOF heat capacities.

F.6.10 MOF water stability

As baselines we use a gradient-boosted classifier and TabPFN on the features reported by Batra et al.¹²⁹ We follow the binary classification setting reported by Batra et al.¹²⁹ in which the kinetically and thermodynamically stable MOFs are merged into one class.

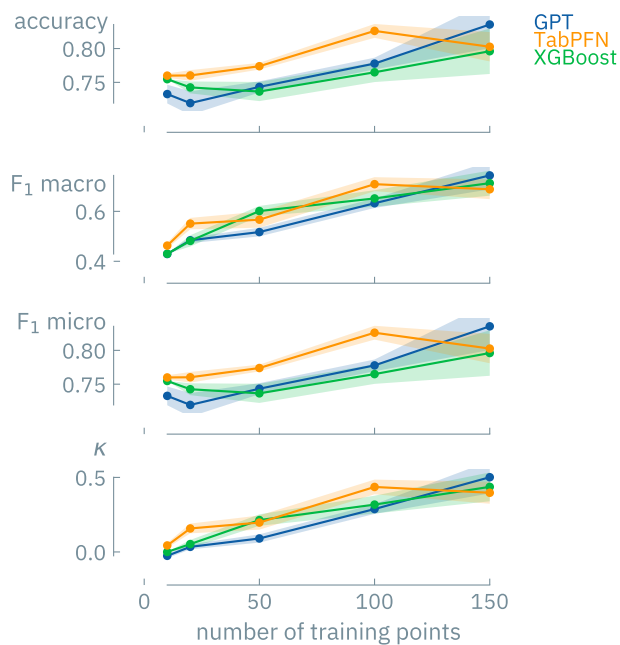


Figure 201: Learning curve for the classification of the water stability of MOFs

F.6.11 Linear polymers

We investigated balanced binary and five-class classification. Figure 202 and Figure 204 show learning curves.

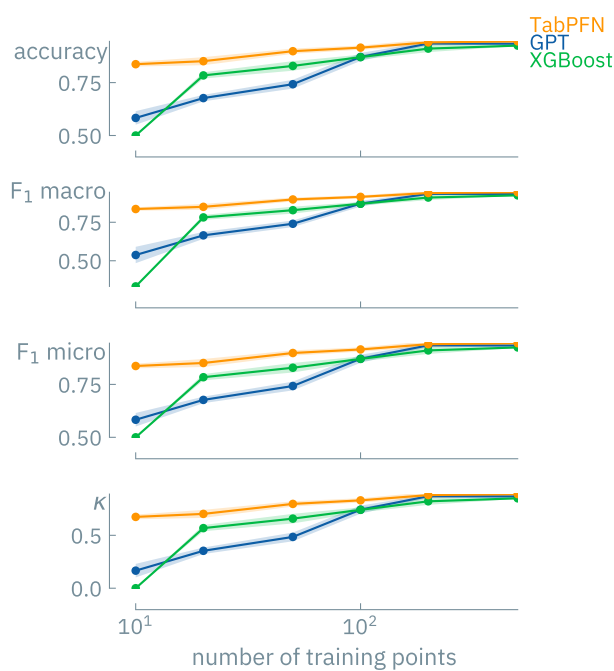


Figure 202: Classification performance for binary classification of free energy of adsorption of linear polymers.

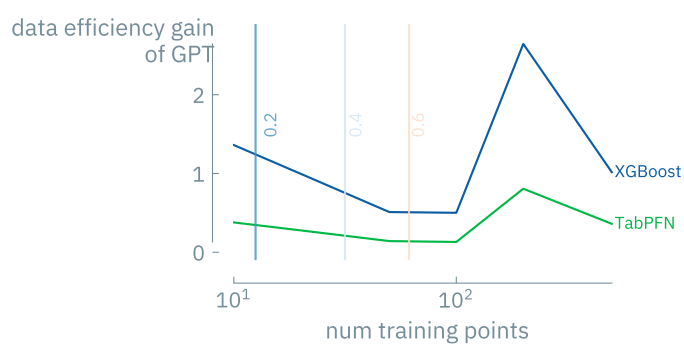


Figure 203: Learning curve intersection points for binary classification on the polymer dataset. Vertical lines indicate the κ scores of the GPT-3 model.

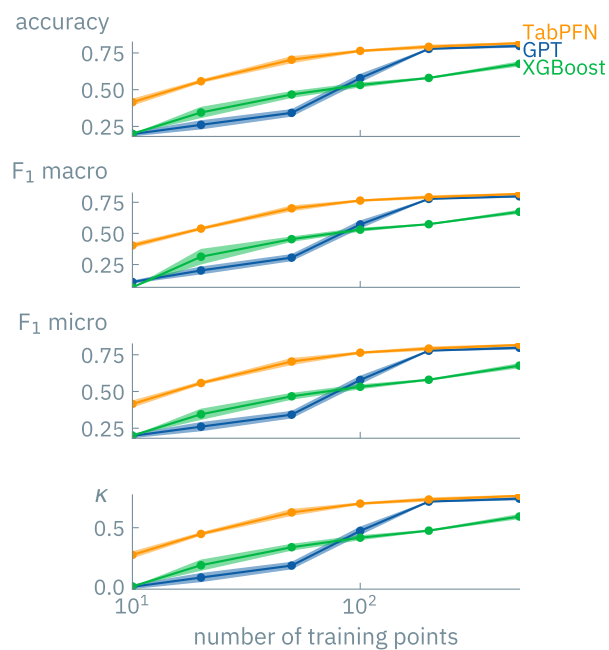


Figure 204: Classification performance for classification of free energy of adsorption of linear polymers into five classes.

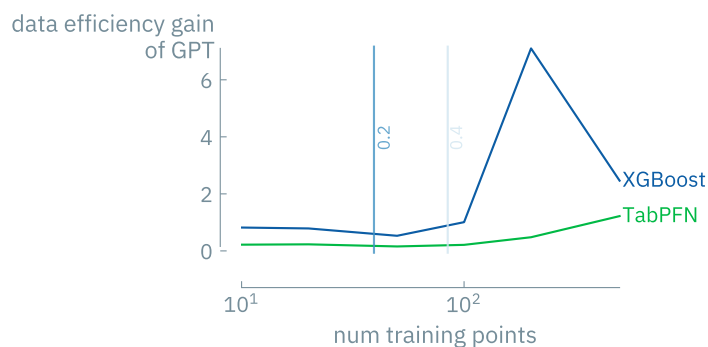


Figure 205: Learning curve intersection points for 5-class classification on the photoswitch dataset. Vertical lines indicate the κ scores of the GPT-3 model.

F.6.12 High-entropy alloys

Single- vs. multi-phase

Unfortunately, Pei et al.⁴⁰⁰ did not report learning curves or a code implementation. However, we find that with very few points and without any feature engineering, we can match their predicted performance. The dataset is balanced. As an alternative baseline, we use automated ML, as implemented in the automatminer package²¹² with the “express” preset.

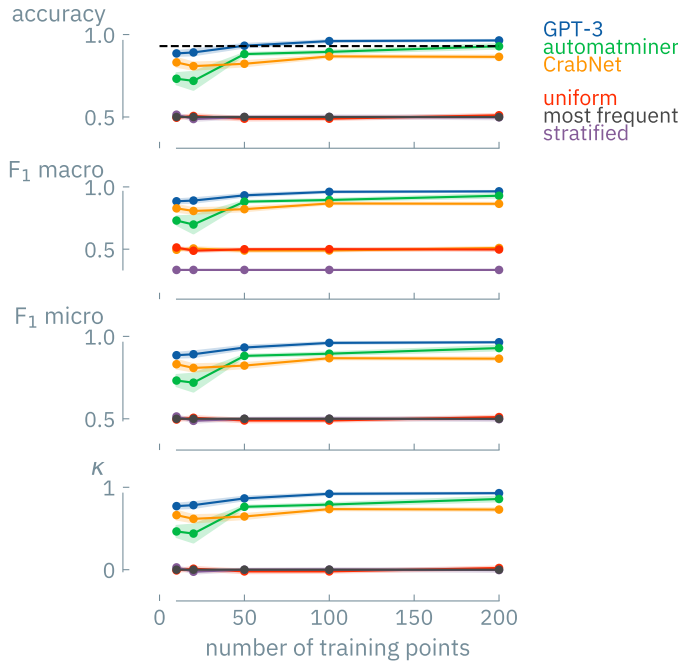


Figure 206: Classification performance for classifying HEAs as “single-phase” and “multi-phase”, respectively. The dashed horizontal line indicates the performance reported in Pei et al.⁴⁰⁰ using a dataset of 1252 points and 10-fold cross-validation, i.e., corresponding to a training set size of around 1126 points. Automatminer baseline computed using “express” preset.

As an additional deep-learning-based baseline, we considered CrabNet⁴⁰¹ if the default model architecture (due to the computational cost of comprehensive hyperparameter optimization and the fact that it performed favorably on matbench on different tasks with the same hyperparameters).

Multiphase, vs. hcp, bcc, fcc

As an extension, we not only considered the binary classification into “single phase” and “multiphase” but directly predicted the structure type (hcp, fcc, bcc) for a given alloy composition. The classes are not balanced, with “multiphase” forming the majority class. As shown in Figure 207, GPT-3 can learn to predict the phase based on only the composition and with very few data points.

We performed these experiments by fine-tuning for eight epochs with a learning rate decay rate of 0.02.

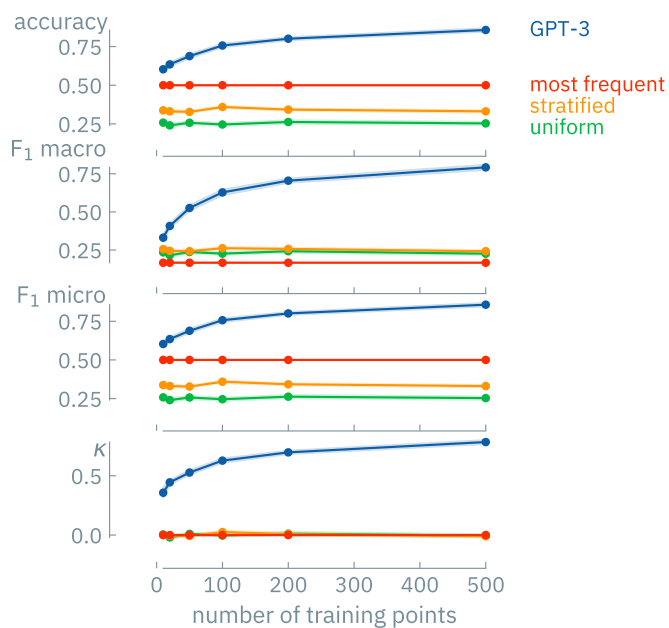


Figure 207: Classification performance for classifying HEAs into multiphase, hcp, fcc, bcc, respectively. Since Pei et al.⁴⁰⁰ did not report this experiment, wherefore we use simply dummy models as baselines. Error bands show the standard deviations for 10 independent train/test splits.

As a baseline, we used automated machine learning via automatminer (and the “express” setting).²¹²

F.6.13 Reactions

For the reaction case studies, we use baselines based on GPRs and natural language processing (NLP)-derived fingerprints as well as one-hot encoding and bag-of-SMILES representation.^{844,845}

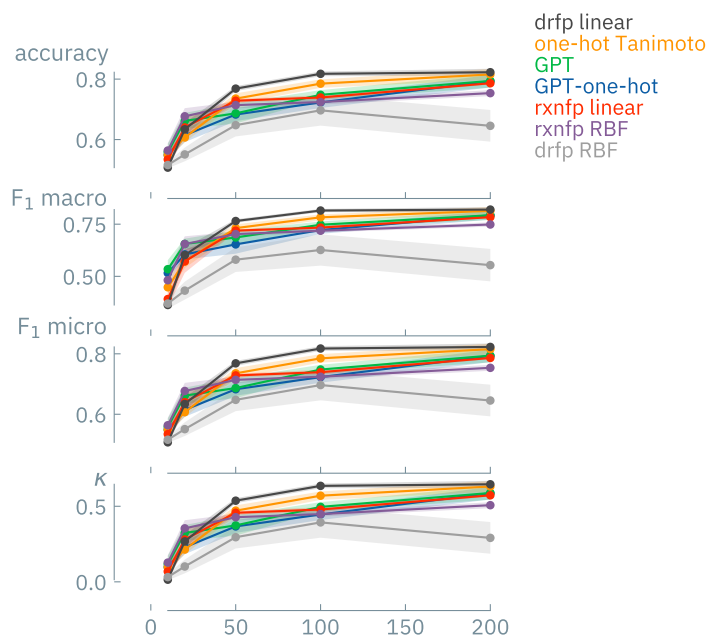
C-N cross-coupling

Figure 208: Learning curves for classification on binary classification of the reaction yield on the data set of Ahneman et al.⁴¹⁸.

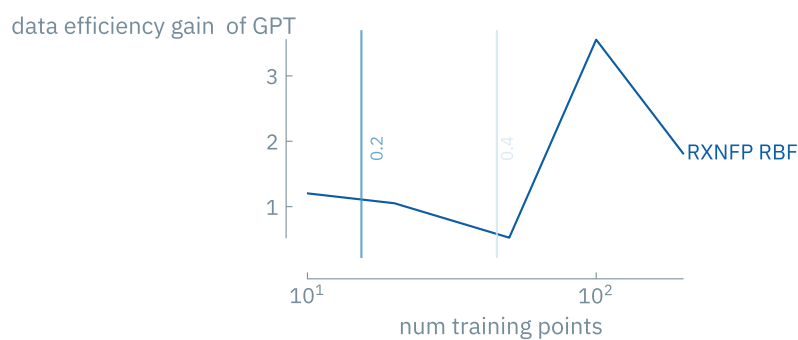


Figure 209: Learning curve intersection points for binary classification on the Ahneman et al.⁴¹⁸ dataset. Vertical lines indicate the κ scores of the GPT-3 model.

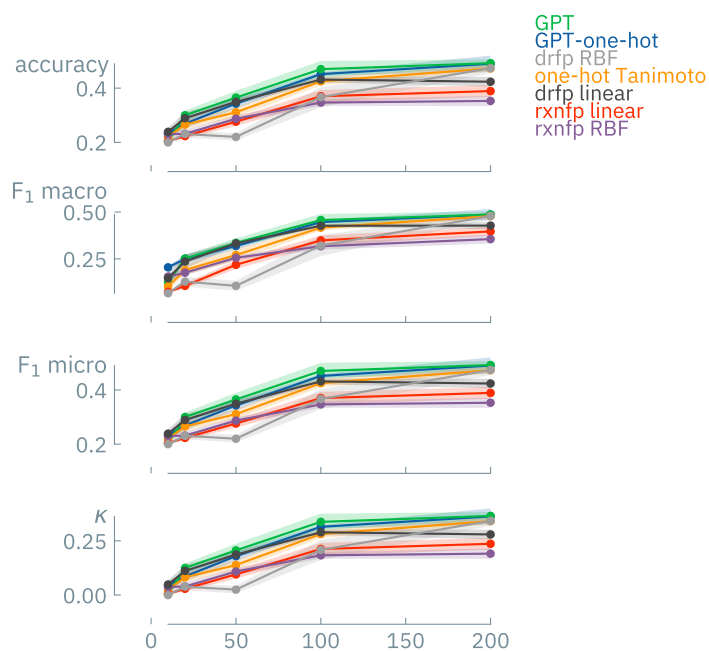


Figure 210: Learning curves for classification on the five-class classification of the reaction yield on the data set of Ahneman et al. ⁴¹⁸.

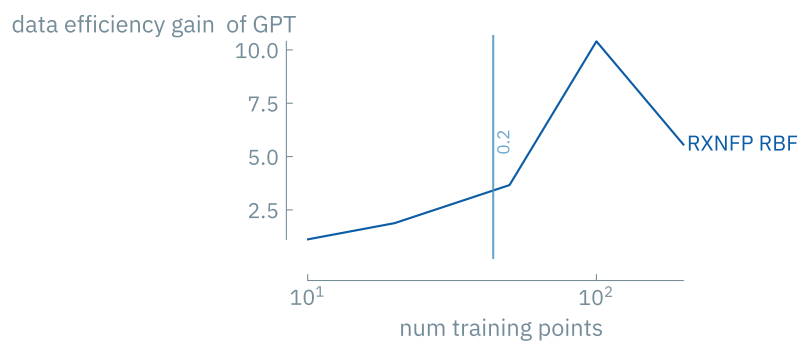


Figure 211: Learning curve intersection points for 5-class classification on the Ahneman et al. ⁴¹⁸ dataset. Vertical lines indicate the κ scores of the GPT-3 model.

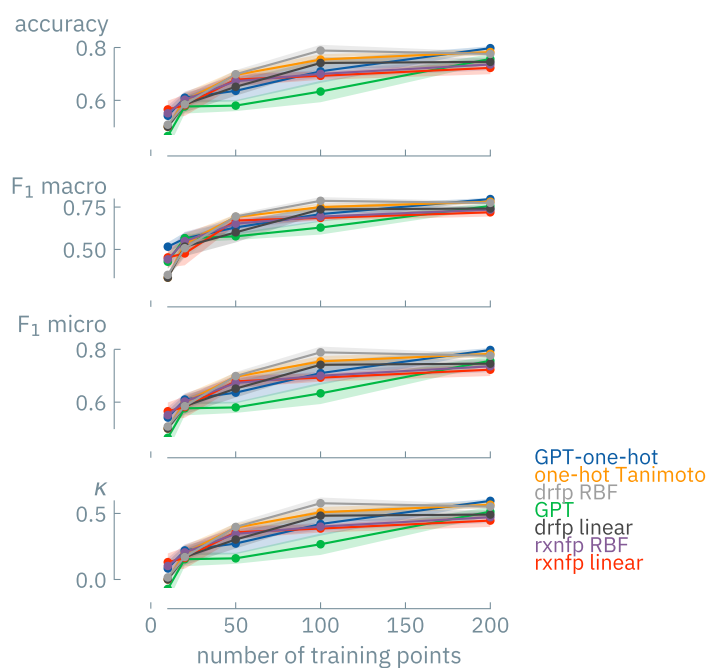
C-C cross-coupling

Figure 212: Learning curves for classification on binary classification of the reaction yield on the data set of Perera et al.⁴¹⁹.

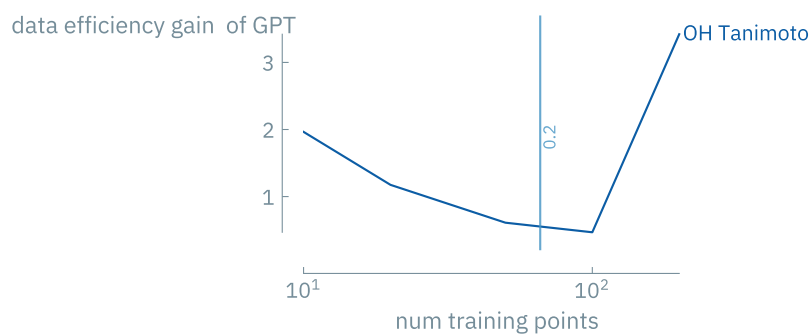


Figure 213: Learning curve intersection points for binary classification on the Perera et al.⁴¹⁹ dataset. Vertical lines indicate the κ scores of the GPT-3 model.

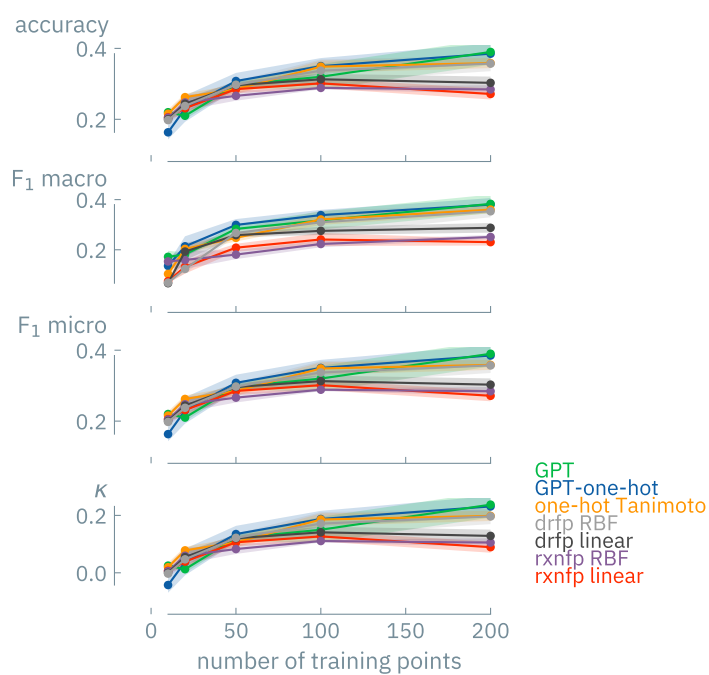


Figure 214: Learning curves for classification on five-class classification of the reaction yield on the data set of Perera et al. ⁴¹⁹.

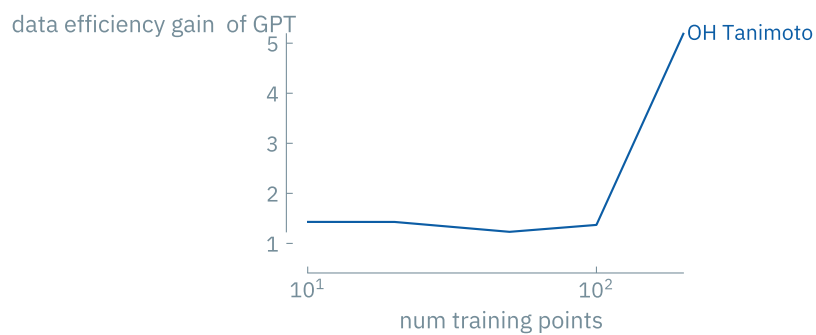
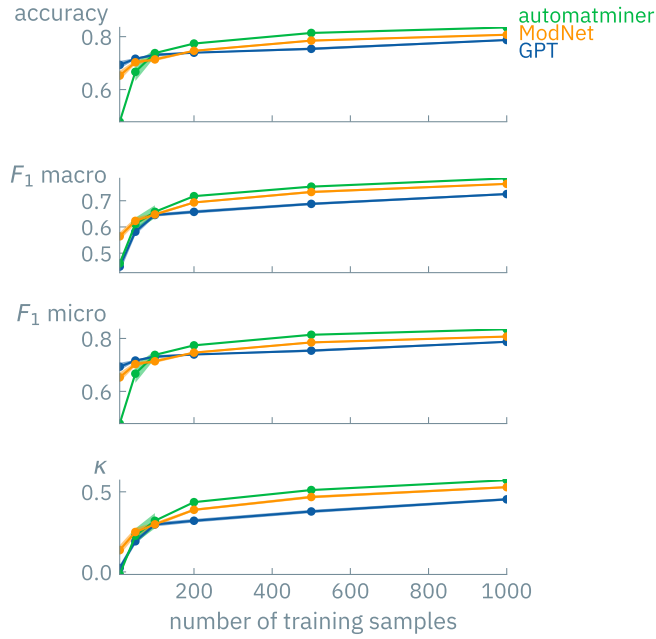


Figure 215: Learning curve intersection points for 5-class classification on the Perera et al. ⁴¹⁹ dataset.

F.6.14 Matbench

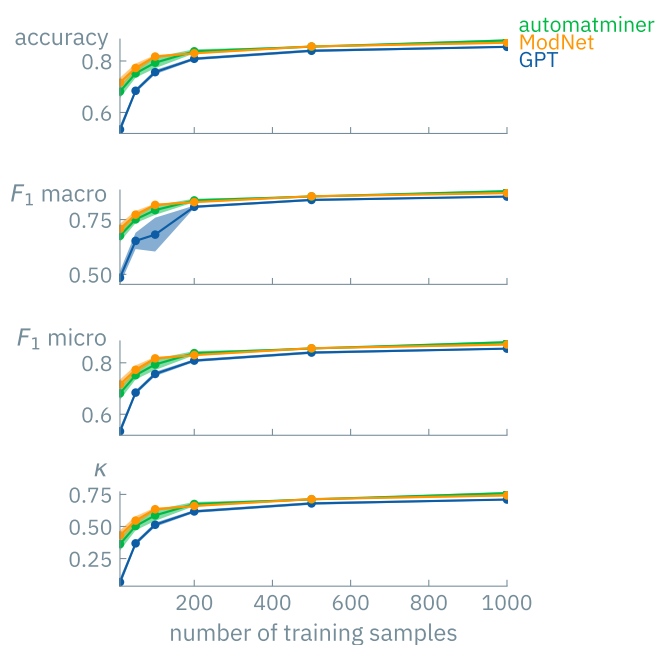
*Metallic glass formation ability***Table 68: Performance of fine-tuning of GPT-3 compared to the current matbench leaderboard for the glass task.**

model	accuracy	balanced accuracy	F_1	ROC-AUC
GPT-3	0.82 ± 0.01	0.77 ± 0.01	0.88 ± 0.01	0.77 ± 0.01
MODNet (v0.1.12) ⁴⁰³	0.97 ± 0.01	0.96 ± 0.01	0.98	0.96 ± 0.01
AMMExpress v2020 ²¹²	0.87 ± 0.05	0.86 ± 0.02	0.90 ± 0.04	0.86 ± 0.02
RF-SCM/Magpie ^{212,615,846,847}	0.90 ± 0.01	0.86 ± 0.02	0.93 ± 0.01	0.86 ± 0.02
MODNet (v0.1.10) ⁴⁰³	0.87 ± 0.01	0.81 ± 0.02	0.91 ± 0.01	0.81 ± 0.02
Dummy	0.59 ± 0.02	0.50 ± 0.02	0.71 ± 0.01	0.50 ± 0.02

**Figure 216: Learning curve analysis for the matbench glass task.** As baselines, we considered the Automatminer with “express” settings and MODNet (with hyperparameters optimized by Breuck et al.⁴⁰³ for this task).

*Metallic behavior***Table 6g: Performance of fine-tuning of GPT-3 compared to the current matbench leader-board for the `expt_is_metal` task.**

model	accuracy	balanced racy	accu- racy	F_1	ROC-AUC
GPT-3	0.89	0.89		0.89	0.89
AMMExpress v2020 ²¹²	0.92	0.92		0.92	0.92
RF- SCM/Magpie ^{212,615,846,847}	0.92 ± 0.01	0.92 ± 0.01		0.92 ± 0.01	0.92 ± 0.01
MODNet (v0.1.10) ⁴⁰³	0.92 ± 0.01	0.92 ± 0.01		0.92 ± 0.01	0.92 ± 0.01
Dummy	0.49 ± 0.01	0.49 ± 0.01		0.49 ± 0.02	0.49 ± 0.01

**Figure 217: Learning curve analysis for the matbench `is_metal` task.** As baselines, we considered the Automatminer with “express” settings and MODNet (with hyperparameters optimized by Breuck et al.⁴⁰³ for this task).

F.7 REGRESSION

F.7.1 Note on regression using GPT-3

Without changes to the architecture and the training procedure, regression, i.e., the prediction of floating point numbers with, in principle, an infinite number of decimal places, is not possible. However, we can approximate regression in two ways. First, one could use many classes, such that the class bins are in the order of the experimental error (often larger than one decimal place). Second, one can try to directly predict rounded floating point numbers.

In both cases, one would expect the performance to be worse than in the classification setting. Here, we focused on the simplest case of directly predicting rounded floating point numbers. Typically, GPT-3 performs worse than baselines in this setting. However, it sometimes approaches the performance of the baselines.

F.7.2 Photoswitches

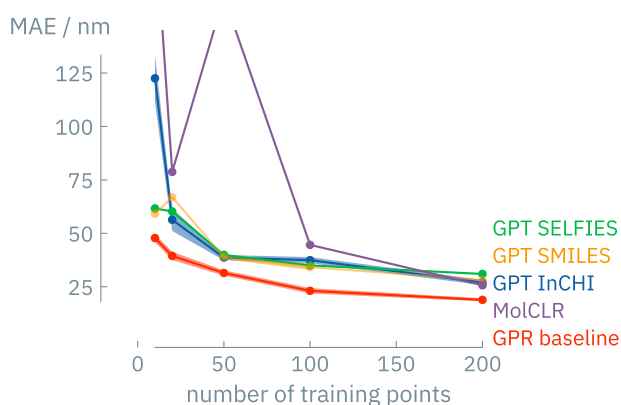


Figure 218: Learning curve for regression on the photoswitch dataset.

F.7.3 HOMO-LUMO gaps

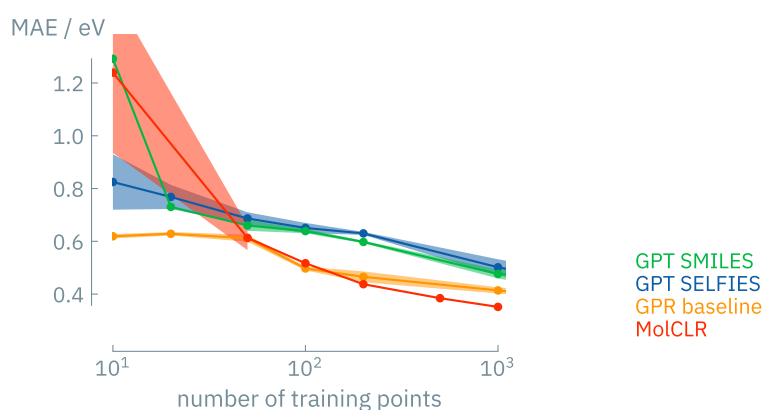


Figure 219: Learning curve for regression on the QMUG dataset.

F.7.4 Solubility

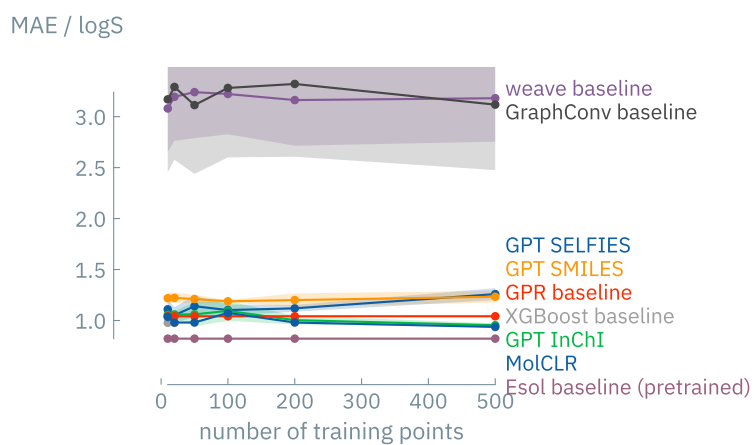


Figure 220: Learning curve for regression on the solubility dataset.

F.7.5 Free energy of solvation



Figure 221: Learning curve for regression on the free energy of solvation dataset.

F.7.6 Lipophilicity

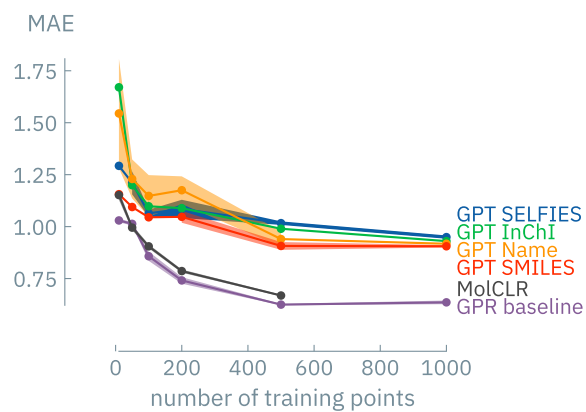


Figure 222: Learning curve for regression on the lipophilicity dataset.

F.7.7 Photoconversion efficiency

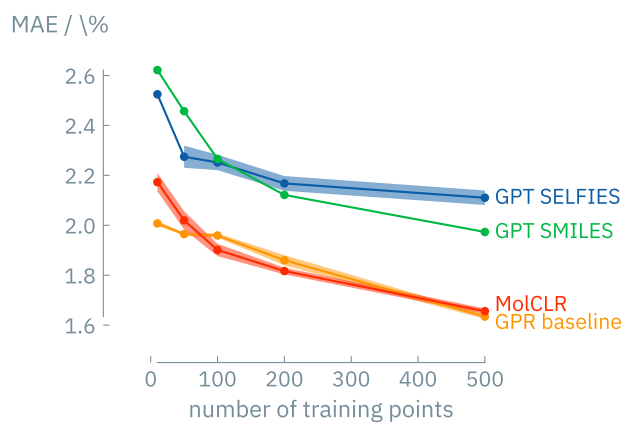


Figure 223: Learning curve for regression on the OPV dataset.

F.7.8 Henry coefficients

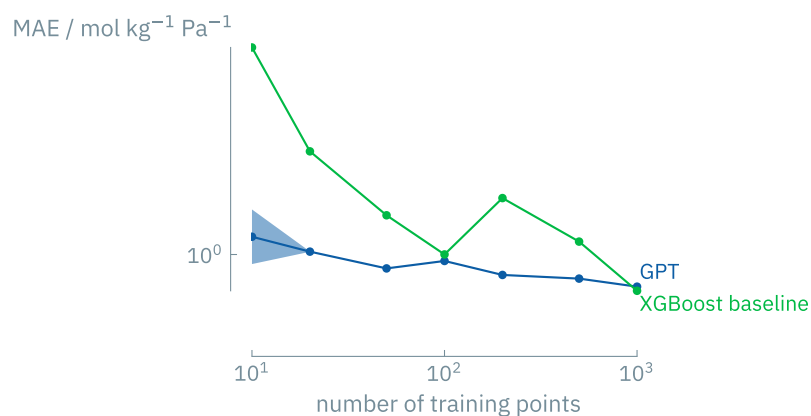
Carbon dioxide

Figure 224: Learning curves for regression for the prediction of the CO_2 Henry coefficient of MOFs.

F.7.9 Surfactants

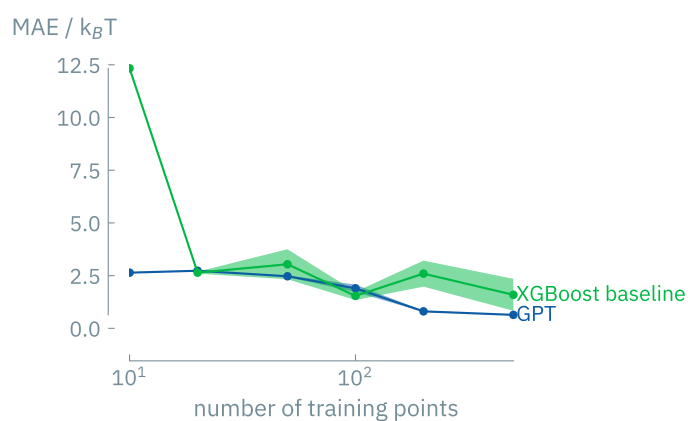


Figure 225: Learning curve for regression for predicting the adsorption free energy on the dispersant dataset.

F.7.10 Reactions

C-N cross-coupling

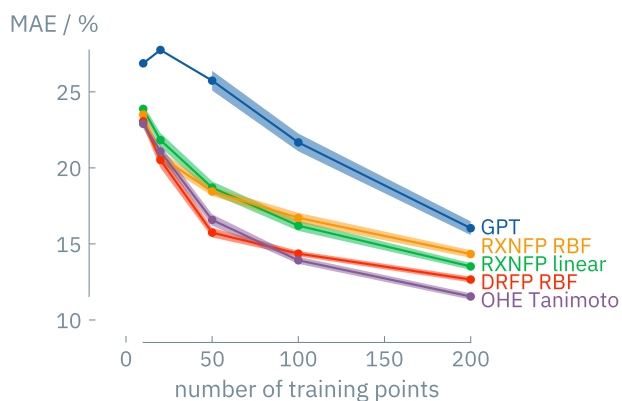


Figure 226: Learning curve for regression yield prediction on the C-N cross-coupling dataset.

C-C cross-coupling

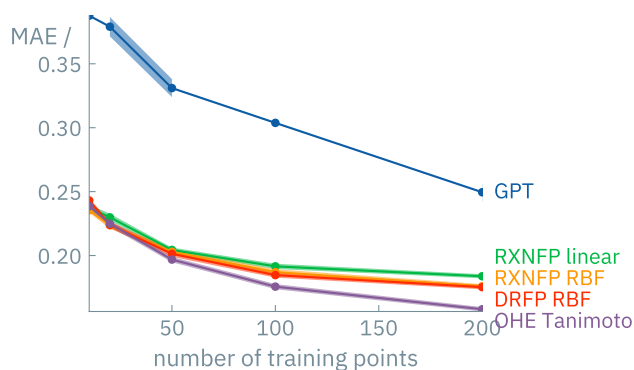


Figure 227: Learning curve for regression yield prediction on the C-C cross-coupling dataset.

F.7.11 Matbench

Also, in the regression setting, fine-tuning of GPT-3 on the composition performs better than the dummy model and is competitive with models on the matbench leaderboard. Table 70 summarizes the metrics for the prediction of band gaps and Table 71 for the prediction of the yield strength of steels.

*Experimental band gaps***Table 70: Performance of a fine-tuned GPT compared to the matbench leaderboard for the expt_gap task.**

model	MAE	RMSE	MAPE	max error
GPT-3	0.46 ± 0.02	1.06 ± 0.09	0.52 ± 0.07	9.36 ± 1.96
Ax SAASBO	0.33 ± 0.01	0.81 ± 0.06	0.36 ± 0.05	7.19 ± 1.97
CrabNet v1.2.7 ^{401,848}				
MODNet (v0.1.12) ⁴⁰³	0.33 ± 0.02	0.77 ± 0.07	0.35 ± 0.04	7.11 ± 1.47
Dummy	1.14 ± 0.03	1.44 ± 0.07	0.95 ± 0.17	8.93 ± 1.23

*Yield strength***Table 71: Performance of a fine-tuned GPT compared to the matbench leaderboard for the steels task.**

model	MAE	RMSE	MAPE	max error
GPT-3	142 ± 16	204 ± 17	0.10 ± 0.01	679 ± 64
MODNet (v0.1.12) ⁴⁰³	88 ± 12	145 ± 37	0.06 ± 0.01	722 ± 277
CrabNet ⁴⁰¹	107 ± 19	153 ± 29	0.07 ± 0.01	477 ± 79
RF-Regex Steels ²¹²	91 ± 7	128 ± 10	0.06 ± 0.01	423 ± 72
Dummy	230 ± 10	301 ± 21	0.16	1032 ± 59

F.8 INVERSE DESIGN

Note that for the inverse design case studies, we focused on examples for which we can readily test the performance of the predicted materials.

F.8.1 Monomer sequences

Since our objectives come from computationally expensive mesoscale simulations, it was not computationally feasible for us to run the simulations for all the generated polymers. Instead, we trained our XGBoost baseline model on all the polymers in our dataset and used this model to score the generated monomer sequences. The performance of this model is shown in Figure 228.

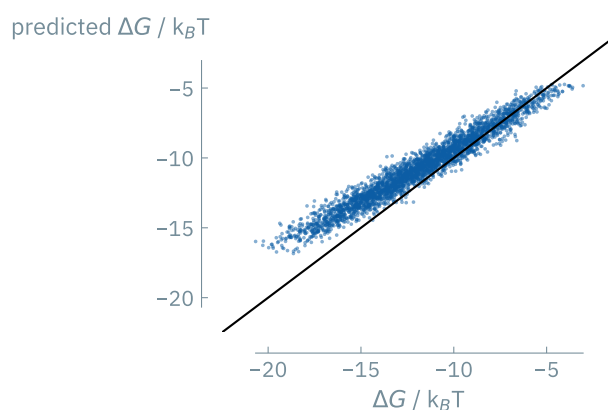


Figure 228: XGBoost model trained on the polymer database. Mean absolute error $1.24 k_B T$.

Random generation continuous target

In Figure 229 we show the performance for generating polymers with properties randomly sampled from the training distribution. We use rounded continuous numbers in the prompts.

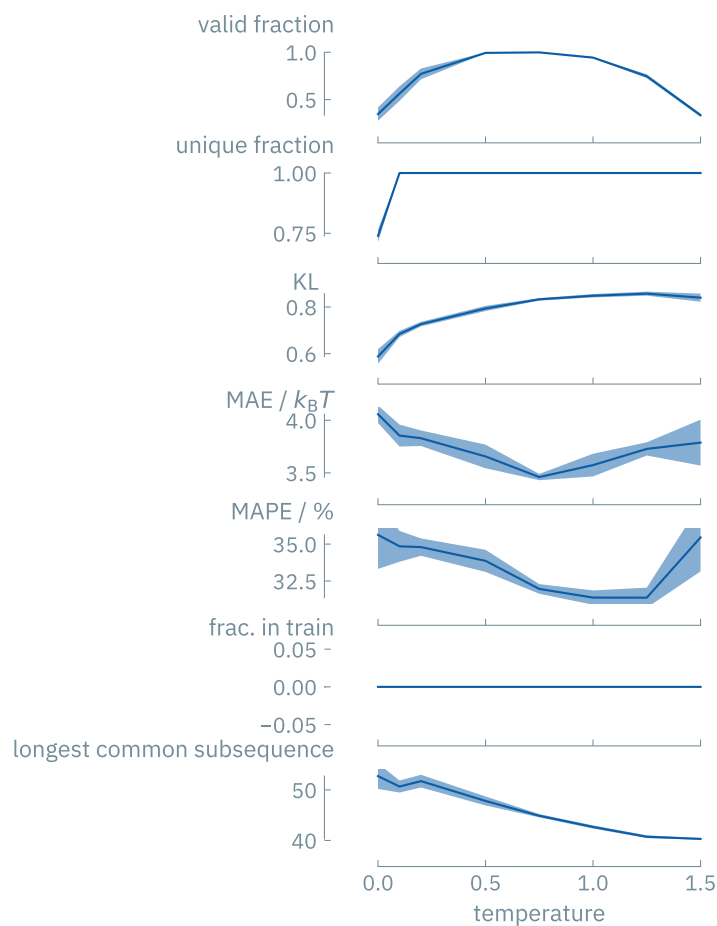


Figure 229: Inverse design metrics for the random generation of polymers as a function of temperature.

Random generation binned target

Instead of using continuous numbers in the prompt, we also considered using the ordinal encoding into five bins. As a loss, we then computed the distance to the nearest bin edge.

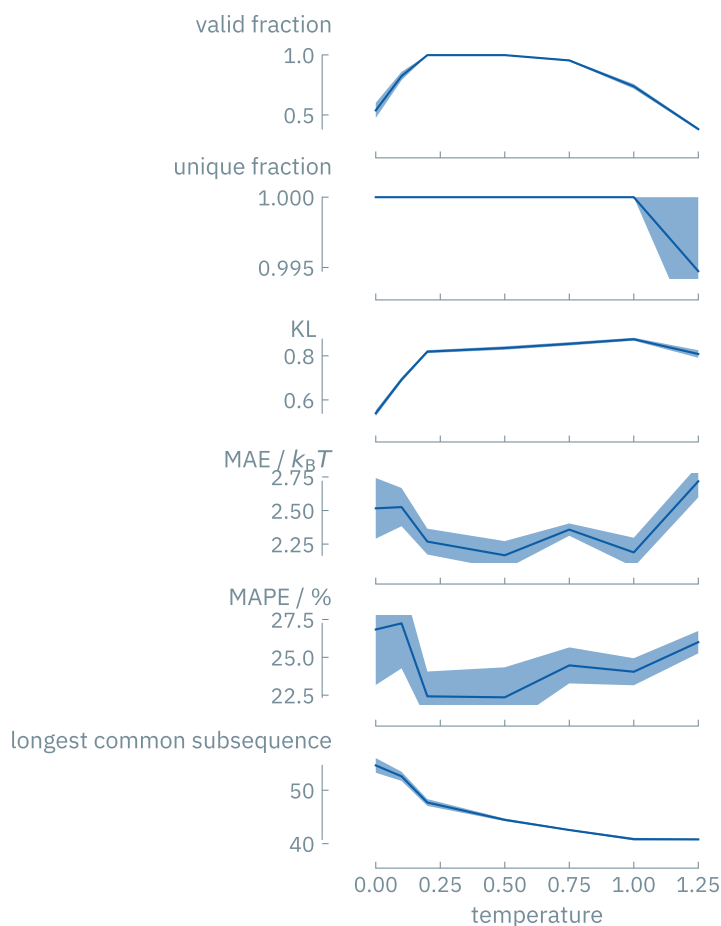


Figure 230: Inverse design metrics for the random generation of polymers as a function of temperature for ordinal design objectives (adsorption free energies).

F.8.2 Photoswitches

To investigate if we can evaluate our generative model using TDDFT, we conducted some experiments to analyze how well we can match the experimental or previously reported DFT data.

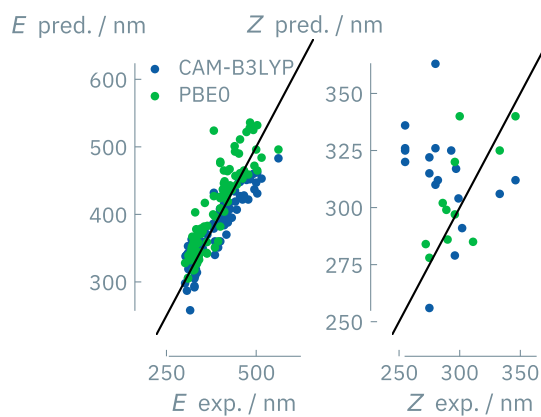


Figure 231: Correlation between experimental transition wavelengths and reported TD-DFT data.

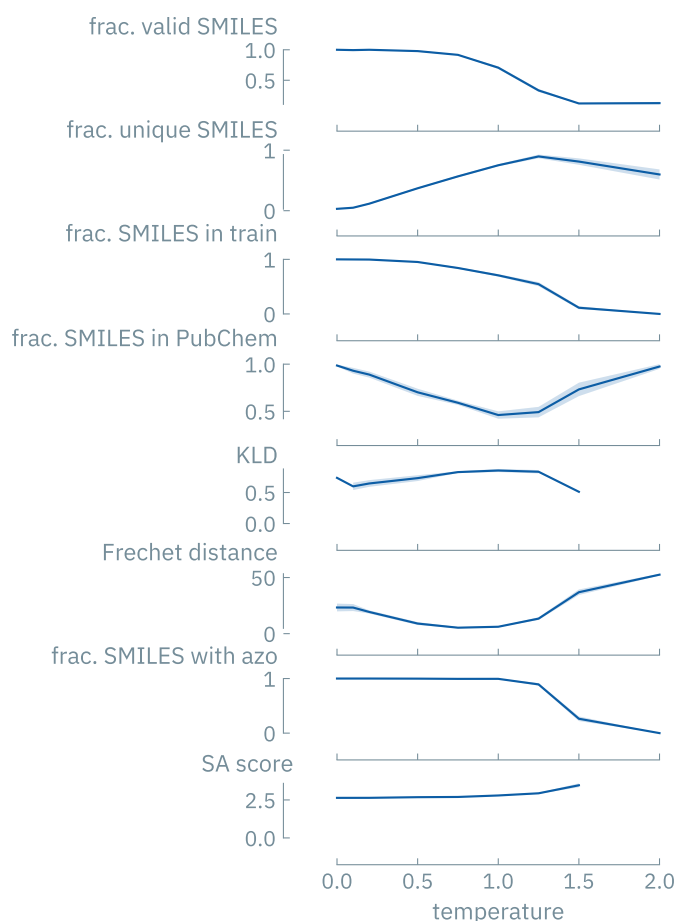


Figure 232: SMILES generation metrics for random generation of photoswitches.

A dummy-mean baseline on the photoswitch dataset, gives a MAE of 53.82 nm for the *E* isomer $\pi - \pi^*$ transition wavelength, and one of 12.63 nm for the *Z* isomer $\pi - \pi^*$ transition wavelength.

Sampling from the training distribution

As the initial experiment, we prompted the model with transition wavelengths randomly sampled from the distribution of the dataset by Griffiths et al.¹⁴³ Figure 232 shows the quality and the diversity of the generated molecules, Figure 233 shows the constrain satisfaction.

Extrapolation

To investigate if our approach can outperform high-throughput virtual screening (HTVS), we only trained our model on photoswitches that adsorb at wavelengths below 350 nm. Upon inference, we queried for photoswitches with transition wavelengths above this threshold.

In Figure 235 we show the quality of the generated SMILES, and in Figure 236 the constrain satisfaction. Figure 237 compares the generated distributions.

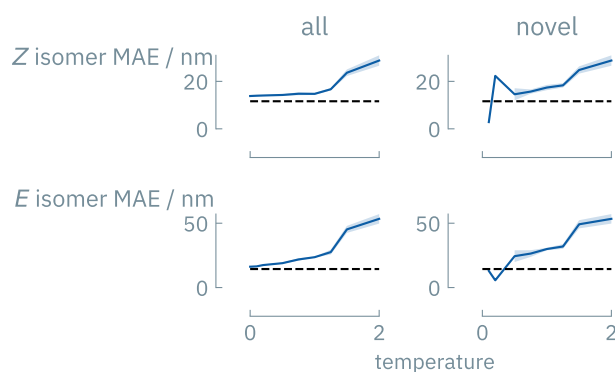


Figure 233: Constrain satisfaction for random generation of photoswitches.

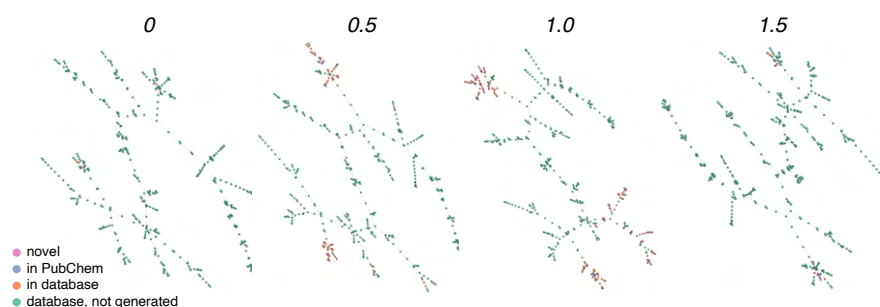


Figure 234: TMAP visualization of the generated photoswitches and the training set. For this visualization, we used the TMAP⁴³⁷ algorithm on photoswitch molecules described using MinHash fingerprint (MHFP) with 2048 permutations.⁴³⁸

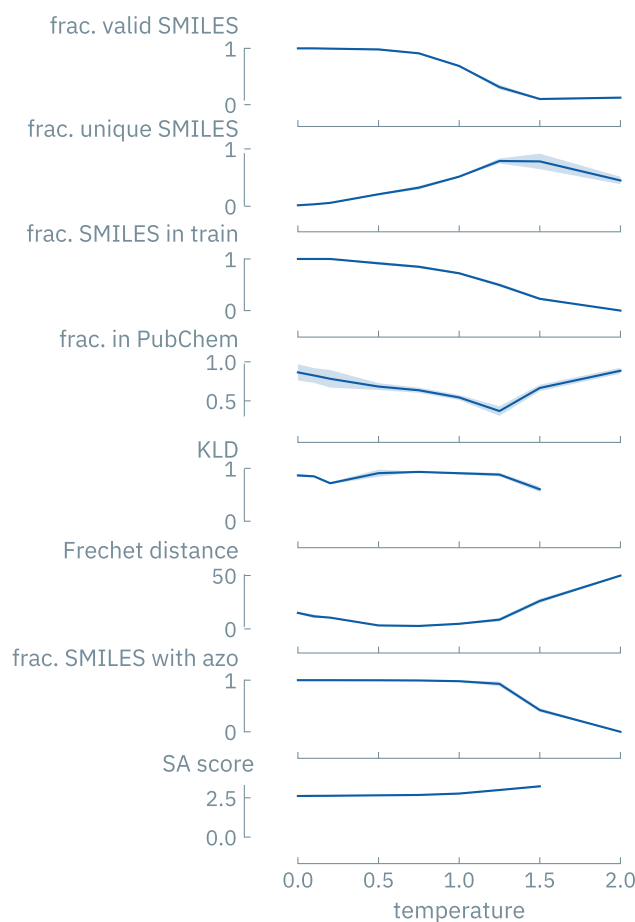


Figure 235: SMILES generation metrics for extrapolative generation of photoswitches. We find that low temperatures generated more valid molecules that, however, are less unique and often part of PubChem.

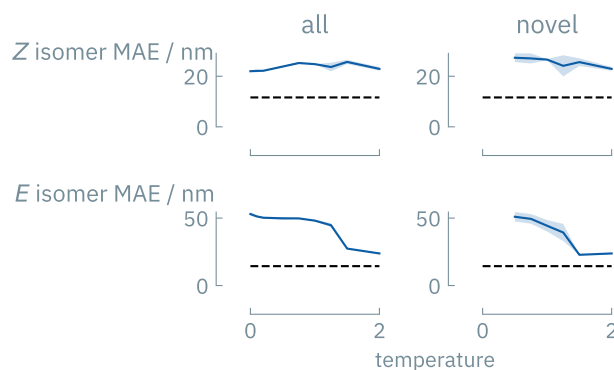


Figure 236: Constrain satisfaction for extrapolative generation of photoswitches. We generally find larger errors than for the random generation. “all” refers to the metrics for all generated photoswitches, including those that have been part of the training set. “novel” only includes those not part of the training set.

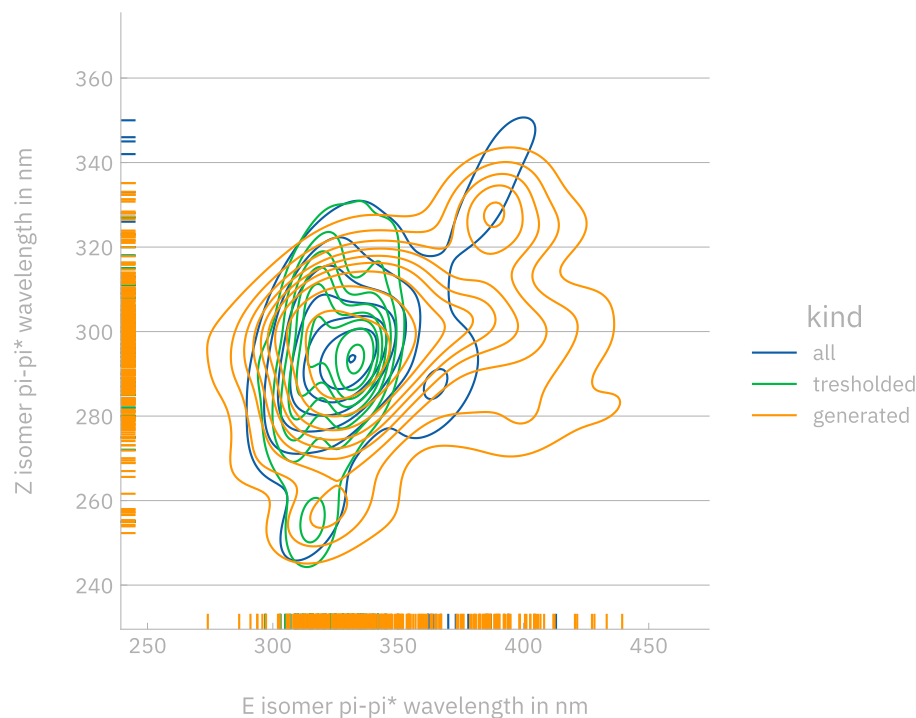


Figure 237: Distribution of transition wavelengths in generated photoswitch molecules, the training set, and the underlying database. For this figure, we considered 192 unique molecules generated across all tested temperatures (0–2) and noise levels. We find that even if we train the model on only a subset of the dataset (green), it can generate molecules (orange) that span a much wider range in transition wavelengths. Transition wavelengths for the generated molecules have been predicted using the GPR models reported by Griffiths et al.¹⁴³

Figure 238 gives TMAP visualizations as a function of temperature.

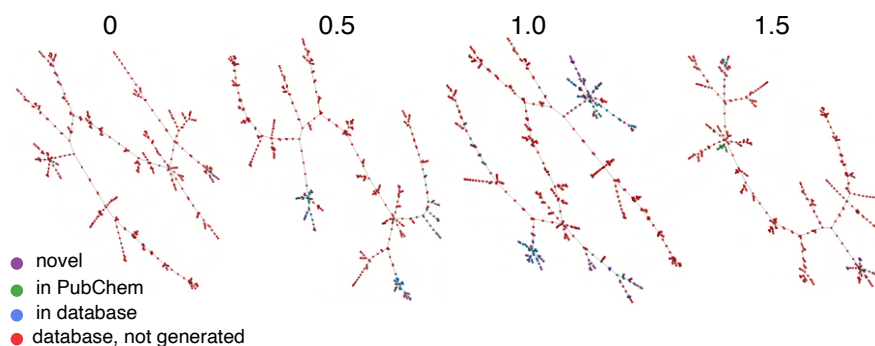


Figure 238: TMAP visualization of the generated photoswitches and the training set. For this visualization, we used the TMAP⁴³⁷ algorithm on photoswitch molecules described using MHFP with 2048 permutations.⁴³⁸ Some molecules generated by GPT-3 do not contain azo groups. Therefore we show in the top row only those that contain azo groups and below all generated valid molecules. We perform the analysis at different inference temperatures (columns).

TDDFT VALIDATION To further analyze our shortlisted candidates, we performed TDDFT simulations. We used Gaussian 16, Revision C.01⁸⁴⁹ to perform geometry optimization followed by the computation of the singlet excited states. Following Jacquemin et al.⁸⁵⁰ we used the PBE0 functional (PBE1PBE)⁸⁵¹ and 6-31G(d',p') basis set,^{852,853} modeling the effect of ethanol solvent using Polarizable Continuum Model (PCM).⁸⁵⁴ Table 72 list the results.

Table 72: TDDFT Predicted transition wavelengths. SMILES generated in the extrapolation experiment (training set containing only molecules with transition wavelengths small than 350 nm).

SMILES	$E \pi-\pi^* / \text{nm}$	$Z \pi-\pi^* / \text{nm}$
<chem>CC1=NOC(C)=C1/N=N/C2=CC=C(NC)C=C2</chem>	386.78	386.79
<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(NC)C=C2</chem>	373.75	387.06
<chem>C[N]1C=CC=C1N=NC2=CC=CC=C2</chem>	371.45	355.75
<chem>CC1=C(C(C)=NN1)/N=N/C2=CC(NC)=CC=C2</chem>	325.23	317.49
<chem>FC1=CC=CC=C1/N=N/C2=CC=C(N)C=C2</chem>	379.68	333.06
<chem>FC1=CC=CC=C1/N=N/C2=C(N)C=CC=C2</chem>	422.21	386.60
<chem>C[N]1C=CC=C1N=NC2=CC(NC)=NN=C2</chem>	393.60	356.72
<chem>C[N]1N=CC(=C1N)N=NC2=CC=CC=C2</chem>	353.73	360.62
<chem>CC1=NOC(C)=C1/N=N/C2=C(N)C=CC=C2</chem>	401.91	376.42
<chem>CC1=C(C(C)=NN1)/N=N/C2=CC(N)C=C2</chem>	341.33	342.74
<chem>C[N]1C=CC=C1N=NC2=CC=CC=C2</chem>	355.75	293.61

F.8.3 HOMO-LUMO gaps

Note that we use HOMO-LUMO gaps computed using GFN2-xTB for computational efficiency. As also reported by Isert et al.⁴¹⁴, there is a deviation between those gaps and those computed using DFT (Figure 239).

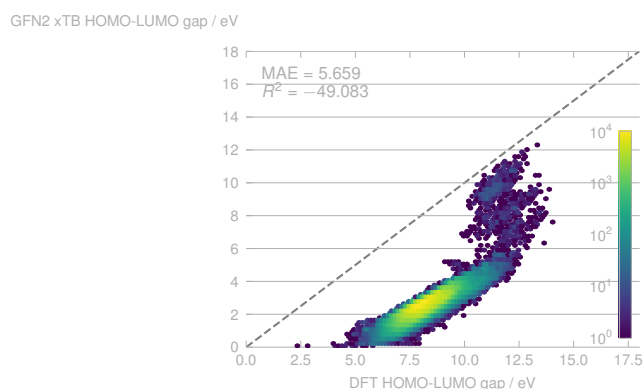


Figure 239: DFT vs. GFN2-xTB bandgaps on the QMUGs dataset.

To measure the compliance of the generated molecules with the prompts, we generated conformers using RDKit (via our `givemeconformer` wrapper⁸⁵⁵ that also performs optimization using the Merck molecular force field,⁸⁵⁶ ranking, and pruning), performed a geometry optimization using GFN2-xTB, and evaluated the HOMO-LUMO gap. Since we did not run metadynamics to generate diverse conformers, our results differ for some cases from the ones reported in QMugs.

On the QMUG dataset, a dummy mean predictor would find a MAE of 0.51 eV.

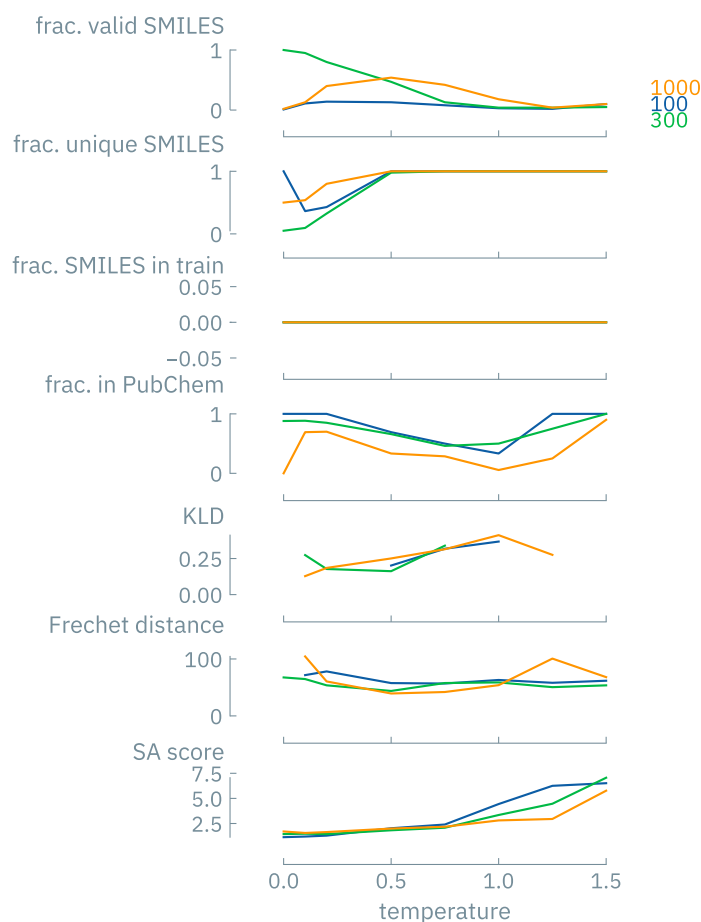


Figure 240: SMILES quality metrics for random generation of SMILES for an inverse model trained on the QMugs dataset. Colors indicate the training set size.

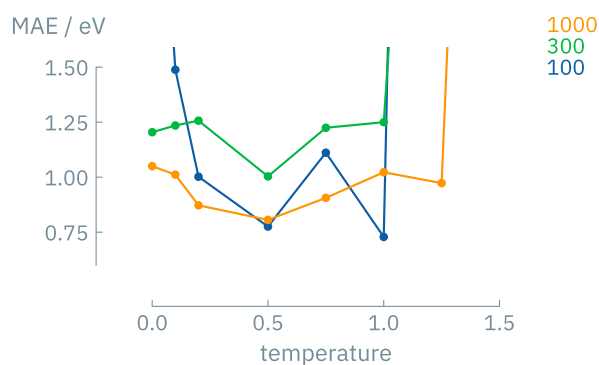


Figure 241: Constrain satisfaction for random generation of SMILES for an inverse model trained on the QMugs dataset. Colors indicate the training set size.

Shorter molecules

Many molecules in the QMugs database have many atoms. To investigate the influence of the SMILES length, we created a subset of the QMugs database in which the SMILES have less than 50 characters.

Precision of the prompt

In our current form of performing inverse design, we create the prompts specifying the desired HOMO-LUMO gap as a floating point number rounded to a specific number of decimal points (by default 2). To investigate the influence of this parameter, we also tested a model in which we rounded to only one decimal point.



Figure 242: SMILES quality metrics for random generation of SMILES for an inverse model trained on the short-SMILES subset QMugs dataset. Colors indicate the training set size.

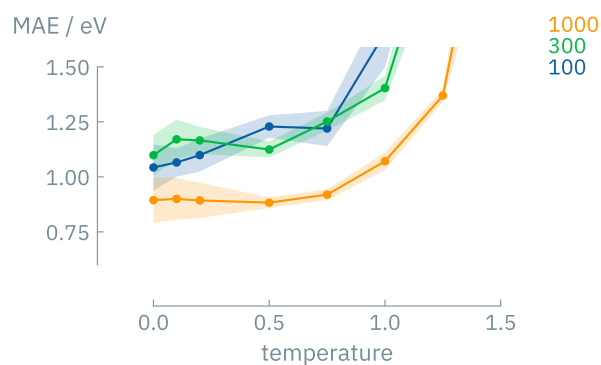


Figure 243: Constraint satisfaction metrics for random generation of SMILES for an inverse model trained on the short-SMILES subset QMugs dataset. Colors indicate the training set size.

Extrapolation

In order for a generative model to be more useful than HTVS, it must be able to generate molecules that lie outside the training distribution. To test GPT-3's abil-

ity to do so, we truncated the QMugs distribution to only include molecules with bandgaps smaller than 3.5 eV.

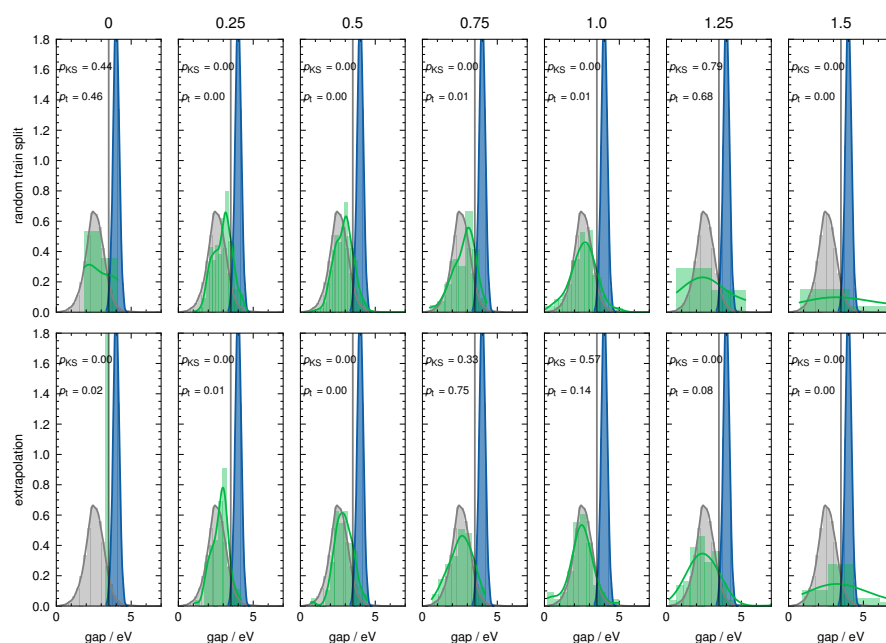


Figure 244: Distribution of HOMO-LUMO gaps of molecules generated by GPT-3 fine-tuned on the random and truncated training set, respectively. The vertical line indicates 3.5 eV, the threshold above which we excluded molecules from the extrapolation training set (bottom row). p -values in the inset are computed using Kolmogorov-Smirnov (p_{KS}) and t -tests (p_t) between the full distribution of HOMO-LUMO gaps of the QMugs database and the one of the generated molecules.

Biasing the generation

To investigate if we can use GPT-3 to shift the generated distribution far from the training distribution, we utilized an iterative approach:

ALGORITHM

Bootstrapping To start the workflow

1. Fine-tune GPT-3 in inverse setting on a random sample
2. Query from a biased distribution, with mean shifted by α with respect to the current mean
3. Evaluate n molecules

Inner loop While the goal is not achieved, do

1. Combine all labeled data
2. Select a training set that is biased toward the target distribution (e.g., truncated with a lower bound)
3. Fine-tune a model

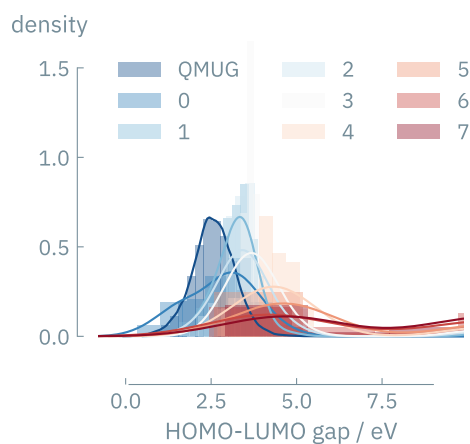


Figure 245: Generated distribution of HOMO-LUMO gaps for biased inverse models. For this experiment, we limited the number of molecules we evaluated using xTB to 100.

4. Query from a biased distribution, with mean shifted by α with respect to the current mean
5. Evaluate n molecules

DIFFERENT BATCH SIZES For the experiment in the main text, we performed the experiment evaluating the following number of evaluations: 2252 in the first, 1997 in the second, 370 in the third, 1875 in the fourth, and 1572 in the last.

Additionally, we also performed the experiment by constraining the number of molecules we evaluate using xTB to 100. Note that we queried many more molecules from the biased distribution and then *randomly* selected 100 from this distribution for evaluation using xTB and we can also successfully shift the distribution (Figure 245). In future work, this can be improved by ranking the candidates using a ML surrogate model.

F.9 PERMUTATION TEST

To test if the model extracted meaningful structure-property relationships, we performed permutation tests (on the classification task on the photoswitch dataset). For this, we randomly shuffled the target column and fine-tuned and tested GPT-3 on this data.

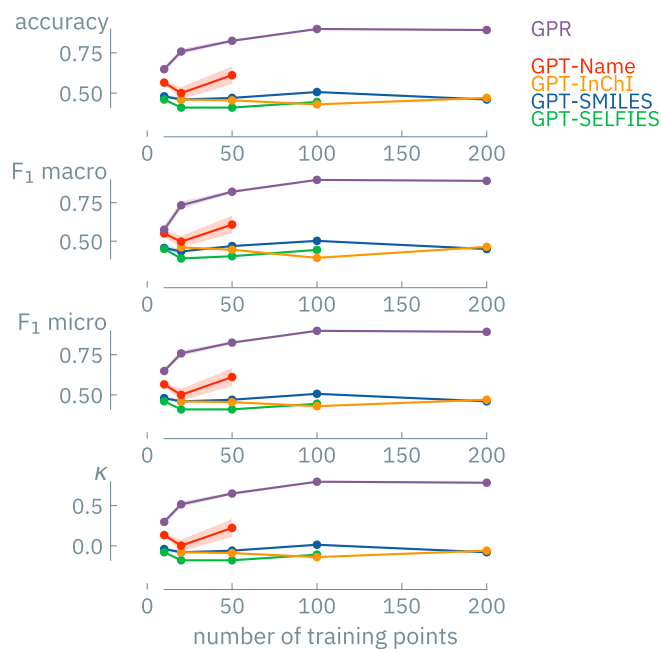


Figure 246: Learning curves for binary classification for GPT-3 models fine-tuned on shuffled versions of the photoswitch dataset. Learning curve for the GPR model trained on unshuffled data shown for reference.

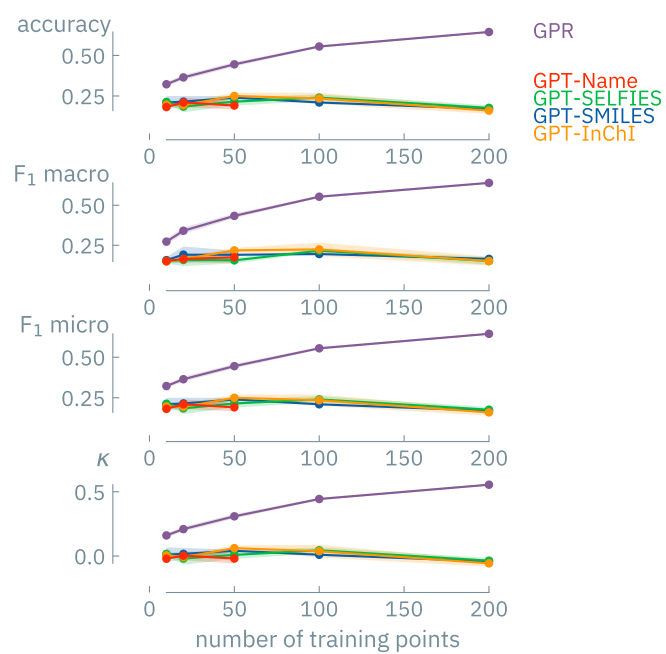


Figure 247: Learning curves for 5-class classification for GPT-3 models fine-tuned on shuffled versions of the photoswitch dataset. Learning curve for the GPR model trained on unshuffled data shown for reference.

F.10 INVALID INPUTS

ML models can confidently hallucinate. To test the behavior of our fine-tuned models, we investigated different inputs: random names, random combinations of elemental symbols as well as random combinations of letters.

F.10.1 Forward classification model

Table 73: Completions for a classification model trained on the photoswitch dataset when fed with inputs that are not a SMILES string.

prompt	completion
What is the transition wavelength of Berend?	0
What is the transition wavelength of Kevin?	0
What is the transition wavelength of Philippe?	0
What is the transition wavelength of Andres?	0
What is the transition wavelength of Bus?	0
What is the transition wavelength of car?	0
What is the transition wavelength of tree?	0
What is the transition wavelength of house?	0
What is the transition wavelength of cat?	0
What is the transition wavelength of magnificent?	0
What is the transition wavelength of a Berend?	0
What is the transition wavelength of a Kevin?	0
What is the transition wavelength of a Philippe?	0
What is the transition wavelength of an Andres?	0
What is the transition wavelength of a Bus?	0
What is the transition wavelength of a car?	0

Table 74: Completions for a classification model trained on the photoswitch dataset when fed with inputs that are not a SMILES string and additional change to the prompt template.

prompt	completion
what is the adsorption energy of Berend?	0
what is the adsorption energy of Kevin?	0
what is the adsorption energy of Philippe?	0
what is the adsorption energy of Andres?	0
what is the adsorption energy of Bus?	0
what is the adsorption energy of car?	0
what is the adsorption energy of tree?	0
what is the adsorption energy of house?	0
what is the adsorption energy of cat?	0
what is the adsorption energy of magnificent?	0
what is the adsorption energy of a Berend?	0
what is the adsorption energy of a Kevin?	0
what is the adsorption energy of a Philippe?	0
what is the adsorption energy of an Andres?	0
what is the adsorption energy of a Bus?	0
what is the adsorption energy of a car?	0

Table 75: Completions for a classification model trained on the photoswitch dataset when fed with inputs that are not a SMILES string and not even valid words.

prompt	completion
What is the transition wavelength of jtjei?	0
What is the transition wavelength of jytcS?	0
What is the transition wavelength of cymtv?	0
What is the transition wavelength of ntnlz?	0
What is the transition wavelength of oqeze?	0
What is the transition wavelength of linwg?	0
What is the transition wavelength of ktgje?	0
What is the transition wavelength of cjmqi?	0
What is the transition wavelength of srved?	0

Table 76: Completions for a classification model trained on the photoswitch dataset when fed with inputs that are not a SMILES string but a random combination of element symbols.

prompt	completion
What is the transition wavelength of PFeScNaFe?	0
What is the transition wavelength of CIMnCuScBe?	0
What is the transition wavelength of GeClNaCFe?	0
What is the transition wavelength of BeCuGeZnFe?	0
What is the transition wavelength of CoGeNaMnCr?	0
What is the transition wavelength of CaFOGaNi?	0
What is the transition wavelength of NaKNeNiN?	0
What is the transition wavelength of GeNeMgMnMg?	0
What is the transition wavelength of ZnNiZnFF?	0
What is the transition wavelength of CaTiAsCoMg?	0

F.10.2 Inverse design model

The results for invalid inputs to an inverse design model are shown in Table 77–Table 79.

Table 77: Completions for an inverse design model trained on the photoswitch dataset when fed with inputs that are not a SMILES string (but valid words or names).

prompt	completion
What is a molecule with E isomer transition wavelength Berend, Z isomer transition wavelength Berend?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength Kevin, Z isomer transition wavelength Kevin?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength Philippe, Z isomer transition wavelength Philippe?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength Andres, Z isomer transition wavelength Andres?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2</chem>
What is a molecule with E isomer transition wavelength Bus, Z isomer transition wavelength Bus?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(Br)C=C2</chem>
What is a molecule with E isomer transition wavelength car, Z isomer transition wavelength car?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength tree, Z isomer transition wavelength tree?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C)C=C2</chem>
What is a molecule with E isomer transition wavelength house, Z isomer transition wavelength house?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength cat, Z isomer transition wavelength cat?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength magnificent, Z isomer transition wavelength magnificent?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength a Berend, Z isomer transition wavelength a Berend?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength a Kevin, Z isomer transition wavelength a Kevin?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC(NC(C)=O)=CC=C2</chem>
What is a molecule with E isomer transition wavelength a Philippe, Z isomer transition wavelength a Philippe?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength an Andres, Z isomer transition wavelength an Andres?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength a Bus, Z isomer transition wavelength a Bus?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(Br)C=C2</chem>
What is a molecule with E isomer transition wavelength a car, Z isomer transition wavelength a car?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>

Table 78: Completions for an inverse design model trained on the photoswitch dataset when fed with inputs that are not a SMILES string (but random combinations of element symbols).

prompt	completion
What is a molecule with E isomer transition wavelength PFeScNaFe, Z isomer transition wavelength PFeScNaFe?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength ClMnCuScBe, Z isomer transition wavelength ClMnCuScBe?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength GeClNaCFe, Z isomer transition wavelength GeClNaCFe?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength BeCuGeZnFe, Z isomer transition wavelength BeCuGeZnFe?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength CoGeNaMnCr, Z isomer transition wavelength CoGeNaMnCr?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength CaFOGaNi, Z isomer transition wavelength CaFOGaNi?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2</chem>
What is a molecule with E isomer transition wavelength NaKNeNiN, Z isomer transition wavelength NaKNeNiN?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2</chem>
What is a molecule with E isomer transition wavelength GeNeMgMnMg, Z isomer transition wavelength GeNeMgMnMg?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength ZnNiZnFF, Z isomer transition wavelength ZnNiZnFF?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C)C=C2</chem>
What is a molecule with E isomer transition wavelength CaTiAsCoMg, Z isomer transition wavelength CaTiAsCoMg?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2</chem>

Table 79: Completions for an inverse design model trained on the photoswitch dataset when fed with inputs that are not a SMILES string (but random combinations of random letters).

prompt	completion
What is a molecule with E isomer transition wavelength jtjei, Z isomer transition wavelength jtjei?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2</chem>
What is a molecule with E isomer transition wavelength jytcs, Z isomer transition wavelength jytcs?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2</chem>
What is a molecule with E isomer transition wavelength cymtv, Z isomer transition wavelength cymtv?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2</chem>
What is a molecule with E isomer transition wavelength ntnlz, Z isomer transition wavelength ntnlz?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC(C(O)=O)=CC=C2</chem>
What is a molecule with E isomer transition wavelength oqeze, Z isomer transition wavelength oqeze?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2</chem>
What is a molecule with E isomer transition wavelength linwg, Z isomer transition wavelength linwg?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength ktgje, Z isomer transition wavelength ktgje?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2</chem>
What is a molecule with E isomer transition wavelength cjmqi, Z isomer transition wavelength cjmqi?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC(C(O)=O)=CC=C2</chem>
What is a molecule with E isomer transition wavelength srved, Z isomer transition wavelength srved?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2</chem>
What is a molecule with E isomer transition wavelength xjkgf, Z isomer transition wavelength xjkgf?	<chem>CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2</chem>

F.11 NOVEL TASKS

We also investigated how our fine-tuned inverse design models can perform few-shot tasks.

For this, we added statements such as “and F as part of the molecule?” to the end of the prompt and measured if the generated molecules were enriched with the specified functional group. To quantify potential changes in distribution, we randomly sampled from the distribution of properties on the training dataset and then computed the enrichment factor. We queried a model fine-tuned on 1000 randomly sampled molecules from QMugs.

$$\text{enrichment factor} = \frac{\text{fraction in valid SMILES in generated batch}}{\text{fraction in training set}} \quad (40)$$

Note that we do use an indicator function that only considers if a functional group is a molecule or not to compute the fractions. We do not consider the count of a given functional group in a molecule.

BIBLIOGRAPHY

- (1) Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; Sutskever, I. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021*, ed. by Meila, M.; Zhang, T., PMLR: 2021; Vol. 139, pp 8821–8831.
- (2) Smil, V., *Making the modern world*; John Wiley & Sons: Nashville, TN, 2013.
- (3) Titirici, M. et al. *J. Phys. Mater.* **2022**, *5*, 032001.
- (4) Ritchie, H.; Roser, M.; Rosado, P. Energy, <https://ourworldindata.org/energy>, accessed 2023-3-9, 2022.
- (5) Smil, V., *Energy Transitions*, 2nd ed.; Praeger: Westport, CT, 2016.
- (6) bp *Statistical Review of World Energy 2022*; tech. rep. 71st edition; London: bp, 2022.
- (7) Wolfram Research, I. Wolfram | Alpha Knowledgebase, Champaign, IL, 2020.
- (8) Chu, S.; Cui, Y.; Liu, N. *Nat. Mat.* **2016**, *16*, 16–22.
- (9) Ritchie, H.; Roser, M.; Rosado, P. CO₂ and Greenhouse Gas Emissions, <https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions>, accessed 2023-3-9, 2020.
- (10) Masson-Delmotte, V.; Zhai, P.; Pörtner, H.-O.; Roberts, D.; Skea, J.; Shukla, P. R.; Pirani, A.; Moufouma-Okia, W.; Péan, C.; Pidcock, R., et al. *An IPCC Special Report on the impacts of global warming of 2018*, *1*, 43–50.
- (11) Bui, M. et al. *Energ. Environ. Sci.* **2018**, *11*, 1062–1176.
- (12) Anderson, P. W. *Science* **1972**, *177*, 393–396.
- (13) Kalmutzki, M. J.; Hanikel, N.; Yaghi, O. M. *Sci. Adv.* **2018**, *4*, eaat9180.
- (14) Eddaoudi, M.; Kim, J.; Rosi, N.; Vodak, D.; Wachter, J.; O’Keeffe, M.; Yaghi, O. M. *Science* **2002**, *295*, 469–472.
- (15) Wang, Z.; Cohen, S. M. *Chem. Soc. Rev.* **2009**, *38*, 1315.
- (16) Lyu, H.; Ji, Z.; Wuttke, S.; Yaghi, O. M. *Chem* **2020**, *6*, 2219–2241.
- (17) Deng, H.; Doonan, C. J.; Furukawa, H.; Ferreira, R. B.; Towne, J.; Knobler, C. B.; Wang, B.; Yaghi, O. M. *Science* **2010**, *327*, 846–850.
- (18) Xu, W.; Tu, B.; Liu, Q.; Shu, Y.; Liang, C.-C.; Diercks, C. S.; Yaghi, O. M.; Zhang, Y.-B.; Deng, H.; Li, Q. *Nat. Rev. Mater.* **2020**, *5*, 764–779.
- (19) Maine, E.; Garnsey, E. *Res. Policy* **2006**, *35*, 375–393.
- (20) Luna, P. D.; Wei, J.; Bengio, Y.; Aspuru-Guzik, A.; Sargent, E. *Nature* **2017**, *552*, 23–27.
- (21) Walsh, A. *Nat. Chem.* **2015**, *7*, 274–275.
- (22) Walsh, A.; Sokol, A. A.; Buckeridge, J.; Scanlon, D. O.; Catlow, C. R. A. *Nat. Mater.* **2018**, *17*, 958–964.
- (23) Smit, B.; Maesen, T. L. M. *Nature* **2008**, *451*, 671–678.
- (24) Agrawal, A.; Choudhary, A. *APL Mater. Materials* **2016**, *4*, 053208.
- (25) Draxl, C.; Scheffler, M. In *Handbook of Materials Modeling*; Springer International Publishing: 2020, pp 49–73.

- (26) Pietsch, W.; Wernecke, J. In *Berechenbarkeit der Welt?*; Springer Fachmedien Wiesbaden: 2017, pp 37–57.
- (27) Boyd, P. G. et al. *Nature* **2019**, 576, 253–256.
- (28) Nørskov, J. K.; Bligaard, T.; Rossmeisl, J.; Christensen, C. H. *Nat. Chem.* **2009**, 1, 37–46.
- (29) Schaarschmidt, M.; Riviere, M.; Ganose, A. M.; Spencer, J. S.; Gaunt, A. L.; Kirkpatrick, J.; Axelrod, S.; Battaglia, P. W.; Godwin, J. In *arXiv preprint Arxiv-2209.12466*, 2022.
- (30) Jumper, J. et al. *Nature* **2021**, 596, 583–589.
- (31) Shields, B. J.; Stevens, J.; Li, J.; Parasram, M.; Damani, F.; Alvarado, J. I. M.; Janey, J. M.; Adams, R. P.; Doyle, A. G. *Nature* **2021**, 590, 89–96.
- (32) Jia, X.; Lynch, A.; Huang, Y.; Danielson, M.; Lang’at, I.; Milder, A.; Ruby, A. E.; Wang, H.; Friedler, S. A.; Norquist, A. J., et al. *Nature* **2019**, 573, 251–255.
- (33) Raccuglia, P.; Elbert, K. C.; Adler, P. D.; Falk, C.; Wenny, M. B.; Mollo, A.; Zeller, M.; Friedler, S. A.; Schrier, J.; Norquist, A. J. *Nature* **2016**, 533, 73–76.
- (34) Jablonka, K. M.; Moosavi, S. M.; Asgari, M.; Ireland, C.; Patiny, L.; Smit, B. *Chem. Sci.* **2021**, 12, 3587–3598.
- (35) Jablonka, K. M.; Zasso, M.; Patiny, L.; Marzari, N.; Pizzi, G.; Smit, B.; Yakutovich, A. V. In *ChemRxiv preprint 10.26434/chemrxiv-2021-h3381-v2*, 2021.
- (36) Domingues, N. P.; Moosavi, S. M.; Talirz, L.; Jablonka, K. M.; Ireland, C. P.; Ebrahim, F. M.; Smit, B. *Commun. Chem.* **2022**, 5.
- (37) Jablonka, K. M.; Patiny, L.; Smit, B. *J. Chem. Educ.* **2022**, 99, 561–569.
- (38) Jablonka, K. M.; Patiny, L.; Smit, B. *Nat. Chem.* **2022**, 14, 365–376.
- (39) Heidorn, P. B. *Libr. Trends* **2008**, 57, 280–299.
- (40) Baker, M. *Nature* **2016**, 533, 452–454.
- (41) Prinz, F.; Schlange, T.; Asadullah, K. *Nat. Rev. Drug Discovery* **2011**, 10, 712–712.
- (42) Wilkinson, M. D. et al. *Sci. Data* **2016**, 3, 160018.
- (43) Hunter, M., *Establishing the new science: the experience of the early Royal Society*; Boydell Press: Woodbridge, Suffolk England Wolfboro, N.H., USA, 1989.
- (44) McAlpine, J. B. et al. *Nat. Prod. Rep.* **2019**, 36, 35–107.
- (45) Helliwell, J. R.; McMahon, B.; Guss, J. M.; Kroon-Batenburg, L. M. J. *IUCr* **2017**, 4, 714–722.
- (46) CIO Council Support Connecting Americans to Coronavirus Information Online, <https://www.cio.gov/>, accessed 2023-3-9, 2020.
- (47) Google COVID-19 Announcements Structured Data | Google Search Central, <https://developers.google.com/search/docs/advanced/structured-data/special-announcements>, accessed 2023-3-9, 2021.
- (48) Gray, A. J.; Goble, C. A.; Jimenez, R., et al. In *International Semantic Web Conference (Posters, Demos & Industry Tracks)*, 2017.
- (49) Fletcher, G.; Groth, P.; Sequeda, J. In *arXiv preprint Arxiv-2004.07917*, 2020.
- (50) Sporny, M.; Longley, D.; Kellogg, D.; Lanthaler, M.; Champin, P.-A.; Lindström, N. JSON-LD 1.1, ed. by Kellogg, G.; Champin, P.-A.; Longley, D., 2020.

- (51) Group, W. W. *CSV on the Web: A Primer*, ed. by Tennison, J., 2016.
- (52) Coles, S. J.; Frey, J. G.; Bird, C. L.; Whitby, R. J.; Day, A. E. *J. Cheminf.* **2013**, *5*, 52.
- (53) Tremouilhac, P.; Huang, P.-C.; Lin, C.-L.; Huang, Y.-C.; Nguyen, A.; Jung, N.; Bach, F.; Bräse, S. *Chemistry-Methods* **2020**, *1*, 8–11.
- (54) Kwok, R. *Nature* **2018**, *560*, 269–270.
- (55) Kanza, S.; Willoughby, C.; Gibbins, N.; Whitby, R.; Frey, J. G.; Erjavec, J.; Zupančič, K.; Hren, M.; Kovač, K. *J. Cheminf.* **2017**, *9*.
- (56) Rubacha, M.; Rattan, A. K.; Hosselet, S. C. *J. Lab. Autom.* **2011**, *16*, 90–98.
- (57) Guerrero, S.; Dujardin, G.; Cabrera-Andrade, A.; Paz-y-Miño, C.; Indacochea, A.; Inglés-Ferrándiz, M.; Nadimpalli, H. P.; Collu, N.; Dublanche, Y.; Mingo, I. D.; Camargo, D. *PLOS One* **2016**, *11*, ed. by Martens, L., e0160428.
- (58) Dirnagl, U.; Przesdzin, I. *F1000Research* **2016**, *5*, 2.
- (59) Tremouilhac, P.; Nguyen, A.; Huang, Y.-C.; Kotov, S.; Lütjohann, D. S.; Hüb-sch, F.; Jung, N.; Bräse, S. *J. Cheminf.* **2017**, *9*, 54.
- (60) Huang, Y.-C.; Tremouilhac, P.; Nguyen, A.; Jung, N.; Bräse, S. *J. Cheminf.* **2021**, *13*.
- (61) Barillari, C.; Ottoz, D. S.; Fuentes-Serna, J. M.; Ramakrishnan, C.; Rinn, B.; Rudolf, F. *Method. Biochem. Anal.* **2016**, *32*, 638–640.
- (62) Patiny, L.; Zasso, M.; Kostro, D.; Bernal, A.; Castillo, A. M.; Bolaños, A.; Asencio, M. A.; Pellet, N.; Todd, M.; Schloerer, N.; Kuhn, S.; Holmes, E.; Javor, S.; Wist, J. *Magn. Reson. Chem.* **2017**, *56*, 520–528.
- (63) A. Badiola, K. et al. *Chem. Sci.* **2015**, *6*, 1614–1629.
- (64) Williamson, A. E. et al. *ACS Centr. Sci.* **2016**, *2*, 687–701.
- (65) Woelfle, M.; Olliaro, P.; Todd, M. H. *Nat. Chem.* **2011**, *3*, 745–748.
- (66) Carpi, N.; Minges, A.; Piel, M. *J. Open Source Softw.* **2017**, *2*, 146.
- (67) Rudolphi, F. *Nachr. Chem.* **2010**, *58*, 548–550.
- (68) Brandt, N.; Griem, L.; Herrmann, C.; Schoof, E.; Tosato, G.; Zhao, Y.; Zschumme, P.; Selzer, M. *Data Sci. J.* **2021**, *20*.
- (69) Coley, C. W. In *Artificial Intelligence in Drug Discovery*; Royal Society of Chemistry: 2020, pp 327–348.
- (70) Moosavi, S. M.; Chidambaram, A.; Talirz, L.; Haranczyk, M.; Stylianou, K. C.; Smit, B. *Nat. Commun.* **2019**, *10*.
- (71) Ojea-Jiménez, I.; Bastús, N. G.; Puentes, V. *J. Phys. Chem. C* **2011**, *115*, 15752–15757.
- (72) Huang, Y.; Wang, Z.; Fang, C.; Liu, W.; Lou, X.; Liu, J. *RSC Adv.* **2016**, *6*, 70271–70276.
- (73) Lowe, D. M. Extraction of chemical structures and reactions from the literature, Ph.D. Thesis, University of Cambridge, 2012.
- (74) Jin, W.; Coley, C. W.; Barzilay, R.; Jaakkola, T. S. *CoRR* **2017**, *abs/1709.04555*.
- (75) Kim, E.; Huang, K.; Kononova, O.; Ceder, G.; Olivetti, E. *Matter* **2019**, *1*, 8–12.
- (76) Roughley, S. D.; Jordan, A. M. *J. Med. Chem.* **2011**, *54*, 3451–3479.
- (77) Schneider, N.; Lowe, D. M.; Sayle, R. A.; Tarselli, M. A.; Landrum, G. A. *J. Med. Chem.* **2016**, *59*, 4385–4402.

- (78) Brown, D. G.; Gagnon, M. M.; Boström, J. *J. Med. Chem.* **2015**, *58*, 2390–2405.
- (79) Brown, D. G.; Boström, J. *J. Med. Chem.* **2015**, *59*, 4443–4458.
- (80) L. Bird, C.; Willoughby, C.; G. Frey, J. *Chem. Soc. Rev.* **2013**, *42*, 8157–8175.
- (81) Oleksik, G.; Milic-Frayling, N.; Jones, R. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing - CSCW '14*, ACM Press: 2014.
- (82) Lütjohann, D. S.; Jung, N.; Bräse, S. *Chemometr. Intell. Lab.* **2015**, *144*, 100–107.
- (83) McDonald, R. S.; Wilks, P. A. *Appl. Spectrosc.* **1988**, *42*, 151–162.
- (84) Chalk, S. J. *J. Cheminf.* **2016**, *8*, 55.
- (85) Mehr, S. H. M.; Craven, M.; Leonov, A. I.; Keenan, G.; Cronin, L. *Science* **2020**, *370*, 101–108.
- (86) European Commission. Directorate General for Research and Innovation., *Turning FAIR into reality: final report and action plan from the European Commission expert group on FAIR data*. Publications Office: 2018.
- (87) Harrow, I.; Balakrishnan, R.; Jimenez-Ruiz, E.; Jupp, S.; Lomax, J.; Reed, J.; Romacker, M.; Senger, C.; Splendiani, A.; Wilson, J.; Woollard, P. *Drug Discov. Today* **2019**, *24*, 2068–2075.
- (88) Davies, A.; Patiny, L. *Spectrosc. Eur.* **2021**, *21*.
- (89) Bonney, R.; Shirk, J.; Phillips, T.; Wiggins, A.; Ballard, H.; Miller-Rushing, A.; Parrish, J. *Science* **2014**, *343*, 1436–1437.
- (90) Nielsen, M., *Reinventing discovery : the new era of networked science*; Princeton University Press: Princeton, N.J, 2012.
- (91) European Organization For Nuclear Research and OpenAIRE Zenodo, <https://www.zenodo.org/>, accessed 2023-3-9, 2013.
- (92) Coudert, F.-X. *Nat. Chem.* **2020**, *12*, 499–502.
- (93) Bradley, J.-C. *Nat. Preced.* **2007**, 10.1038/npre.2007.39.1.
- (94) Jablonka, K. M.; Ongari, D.; Moosavi, S. M.; Smit, B. *Chem. Rev.* **2020**, *120*, 8066–8129.
- (95) Olson, M., *The logic of collective action; public goods and the theory of groups*; Schocken Books: New York, 1971.
- (96) Strasser, B. *Science* **2008**, *322*, 537–538.
- (97) Chodera, J.; Lee, A. A.; London, N.; von Delft, F. *Nat. Chem.* **2020**, *12*, 581–581.
- (98) Perkmann, M.; Schildt, H. *Res. Policy* **2015**, *44*, 1133–1143.
- (99) Jones, M. M.; Chataway, J. *Technol. Anal. Strateg.* **2021**, *33*, 296–306.
- (100) Edwards, A. M.; Bountra, C.; Kerr, D. J.; Willson, T. M. *Nat. Chem. Biol.* **2009**, *5*, 436–440.
- (101) Jung, N.; Deckers, A.; Bräse, S. *Biospektrum* **2017**, *23*, 212.
- (102) Herres-Pawlis, S.; Koepler, O.; Steinbeck, C. *Angew. Chem. Int. Ed.* **2019**, *58*, 10766–10768.
- (103) Steinbeck, C. et al. *Research Ideas and Outcomes* **2020**, *6*, e55852.
- (104) Wulf, C.; Beller, M.; Boenisch, T.; Deutschmann, O.; Hanf, S.; Kockmann, N.; Kraehnert, R.; Oezaslan, M.; Palkovits, S.; Schimmler, S.; Schunk, S. A.; Wagemann, K.; Linke, D. *ChemCatChem* **2021**, *13*, 3223–3236.

- (105) Cooper, D.; Springer, R. *Data Communities: A New Model for Supporting STEM Data Sharing*; tech. rep.; 2019.
- (106) Evans, J. D.; Bon, V.; Senkovska, I.; Kaskel, S. *Langmuir* **2021**, *37*, 4222–4226.
- (107) Siderius, D. NIST/ARPA-E Database of Novel and Emerging Adsorbent Materials, <https://adsorption.nist.gov/isodb/>, 2020.
- (108) Watson, M. *Genome Biol.* **2015**, *16*, 101.
- (109) Tennant, J. Open Science: Just science done right? <https://doi.org/10.6084/m9.figshare.9759353.v1>, accessed 2023-3-9, 2019.
- (110) Long, M.; Schonfeld, R. *Supporting the Changing Research Practices of Chemists*; tech. rep.; 2015.
- (111) Chung, Y. G.; Haldoupis, E.; Bucior, B. J.; Haranczyk, M.; Lee, S.; Zhang, H.; Vogiatzis, K. D.; Milisavljevic, M.; Ling, S.; Camp, J. S.; Slater, B.; Siepmann, J. I.; Sholl, D. S.; Snurr, R. Q. *J. Chem. Eng. Data* **2019**, *64*, 5985–5998.
- (112) Gražulis, S.; Chateigner, D.; Downs, R. T.; Yokochi, A.; Quirós, M.; Lutterotti, L.; Manakova, E.; Butkus, J.; Moeck, P.; Le Bail, A. *J. Appl. Crystallogr.* **2009**, *42*, 726–729.
- (113) Gražulis, S.; Daškevič, A.; Merkys, A.; Chateigner, D.; Lutterotti, L.; Quirós, M.; Serebryanaya, N. R.; Moeck, P.; Downs, R. T.; Le Bail, A. *Nucleic Acids Res.* **2012**, *40*, D420–D427.
- (114) Jablonka, K. M.; Rosen, A. S.; Krishnapriyan, A. S.; Smit, B. *ACS Cent. Sci.* **2023**, 10.1021/acscentsci.2c01177.
- (115) Yaghi, O. M.; Kalmutzki, M. J.; Diercks, C. S., *Introduction to reticular chemistry: metal-organic frameworks and covalent organic frameworks*; John Wiley & Sons: 2019.
- (116) Moghadam, P. Z.; Li, A.; Wiggin, S. B.; Tao, A.; Maloney, A. G. P.; Wood, P. A.; Ward, S. C.; Fairen-Jimenez, D. *Chem. Mater.* **2017**, *29*, 2618–2625.
- (117) Butler, K. T.; Davies, D. W.; Cartwright, H.; Isayev, O.; Walsh, A. *Nature* **2018**, *559*, 547–555.
- (118) Mjolsness, E.; DeCoste, D. *Science* **2001**, *293*, 2051–2055.
- (119) Moosavi, S. M.; Jablonka, K. M.; Smit, B. *J. Am. Chem. Soc.* **2020**, *142*, 20273–20287.
- (120) Rosen, A. S.; Notestein, J. M.; Snurr, R. Q. *Curr Opin Chem Eng* **2022**, *35*, 100760.
- (121) Simon, C. M.; Mercado, R.; Schnell, S. K.; Smit, B.; Haranczyk, M. *Chem. Mater.* **2015**, *27*, 4459–4475.
- (122) Jablonka, K. M.; Ongari, D.; Moosavi, S. M.; Smit, B. *Nat. Chem.* **2021**, *13*, 771–777.
- (123) Rosen, A. S.; Fung, V.; Huck, P.; O'Donnell, C. T.; Horton, M. K.; Truhlar, D. G.; Persson, K. A.; Notestein, J. M.; Snurr, R. Q. *npj Comput. Mater.* **2022**, *8*.
- (124) Rosen, A. S.; Iyer, S. M.; Ray, D.; Yao, Z.; Aspuru-Guzik, A.; Gagliardi, L.; Notestein, J. M.; Snurr, R. Q. *Matter* **2021**, *4*, 1578–1597.
- (125) Moosavi, S. M.; Novotny, B. Á.; Ongari, D.; Moubarak, E.; Asgari, M.; Kadioglu, Ö.; Charalambous, C.; Ortega-Guerrero, A.; Farmahini, A. H.; Sarkisov, L.; Garcia, S.; Noé, F.; Smit, B. *Nat. Mater.* **2022**, *21*, 1419–1425.

- (126) Luo, Y.; Bag, S.; Zaremba, O.; Cierpka, A.; Andreo, J.; Wuttke, S.; Friederich, P.; Tsotsalas, M. *Angew. Chem. Int. Ed.* **2022**, *61*.
- (127) Nandy, A.; Duan, C.; Kulik, H. J. *J. Am. Chem. Soc.* **2021**, *143*, 17535–17547.
- (128) Nandy, A.; Terrones, G.; Arunachalam, N.; Duan, C.; Kastner, D. W.; Kulik, H. J. *Sci. Data* **2022**, *9*.
- (129) Batra, R.; Chen, C.; Evans, T. G.; Walton, K. S.; Ramprasad, R. *Nat. Mach. Intell.* **2020**, *2*, 704–710.
- (130) Kapoor, S.; Narayanan, A. In *arXiv preprint Arxiv-2207.07048*, 2022.
- (131) Stein, H. S. *Trends Chem.* **2022**, *4*, 682–684.
- (132) Moosavi, S. M.; Nandy, A.; Jablonka, K. M.; Ongari, D.; Janet, J. P.; Boyd, P. G.; Lee, Y.; Smit, B.; Kulik, H. J. *Nat. Commun.* **2020**, *11*, 4068.
- (133) Burner, J.; Luo, J.; White, A.; Mirmiran, A.; Kwon, O.; Boyd, P. G.; Maley, S.; Gibaldi, M.; Simrod, S.; Ogden, V.; Woo, T. K. *Chem. Mater.* **2023**, *35*, 900–916.
- (134) Chung, Y. G.; Camp, J.; Haranczyk, M.; Sikora, B. J.; Bury, W.; Krungleviciute, V.; Yildirim, T.; Farha, O. K.; Sholl, D. S.; Snurr, R. Q. *Chem. Mater.* **2014**, *26*, 6185–6192.
- (135) Malik, M. M. In *arXiv preprint Arxiv-2002.05193*, 2020.
- (136) Bender, A.; Schneider, N.; Segler, M.; Walters, W. P.; Engkvist, O.; Rodrigues, T. *Nat. Rev. Chem.* **2022**, *6*, 428–442.
- (137) Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. In *2009 IEEE conference on computer vision and pattern recognition*, 2009, pp 248–255.
- (138) Donoho, D. *J. Comput. Graph. Stat.* **2017**, *26*, 745–766.
- (139) Brown, N.; Fiscato, M.; Segler, M. H.; Vaucher, A. C. *J. Chem. Inf. Model.* **2019**, *59*, 1096–1108.
- (140) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. *Sci. Data* **2014**, *1*.
- (141) Blum, L. C.; Reymond, J.-L. *J. Am. Chem. Soc.* **2009**, *131*, 8732.
- (142) Montavon, G.; Rupp, M.; Gobre, V.; Vazquez-Mayagoitia, A.; Hansen, K.; Tkatchenko, A.; Müller, K.-R.; von Lilienfeld, O. A. *New J. Phys.* **2013**, *15*, 095003.
- (143) Griffiths, R.-R.; Greenfield, J. L.; Thawani, A. R.; Jamasb, A. R.; Moss, H. B.; Bourached, A.; Jones, P.; McCorkindale, W.; Aldrick, A. A.; Fuchter, M. J.; Lee, A. A. *Chem. Sci.* **2022**, *13*, 13541–13551.
- (144) Axelrod, S.; Gómez-Bombarelli, R. *Sci. Data* **2022**, *9*, 185.
- (145) Boyd, P. G.; Woo, T. K. *CrystEngComm* **2016**, *18*, 3777–3792.
- (146) Anderson, R.; Rodgers, J.; Argueta, E.; Biong, A.; Gómez-Gualdrón, D. A. *Chem. Mater.* **2018**, *30*, 6325–6337.
- (147) Ongari, D.; Yakutovich, A. V.; Talirz, L.; Smit, B. *ACS Cent. Sci.* **2019**, *5*, 1663–1675.
- (148) Huck, J. M.; Lin, L.-C.; Berger, A. H.; Shahrak, M. N.; Martin, R. L.; Bhowan, A. S.; Haranczyk, M.; Reuter, K.; Smit, B. *Energy Environ. Sci.* **2014**, *7*, 4132–4146.
- (149) Lin, L.-C.; Berger, A. H.; Martin, R. L.; Kim, J.; Swisher, J. A.; Jariwala, K.; Rycroft, C. H.; Bhowan, A. S.; Deem, M. W.; Haranczyk, M.; Smit, B. *Nat. Mater.* **2012**, *11*, 633–641.

- (150) Groom, C. R.; Bruno, I. J.; Lightfoot, M. P.; Ward, S. C. *Acta Crystallogr. B Struct. Sci. Cryst. Eng. Mater.* **2016**, *72*, 171–179.
- (151) Gražulis, S.; Chateigner, D.; Downs, R. T.; Yokochi, A.; Quirós, M.; Lutterotti, L.; Manakova, E.; Butkus, J.; Moeck, P.; Le Bail, A. *J. Appl. Crystallogr.* **2009**, *42*, 726–729.
- (152) Bucior, B. J.; Rosen, A. S.; Haranczyk, M.; Yao, Z.; Ziebel, M. E.; Farha, O. K.; Hupp, J. T.; Siepmann, J. I.; Aspuru-Guzik, A.; Snurr, R. Q. *Cryst. Growth Des.* **2019**, *19*, 6682–6697.
- (153) Barthel, S.; Alexandrov, E. V.; Proserpio, D. M.; Smit, B. *Cryst. Growth Des.* **2018**, *18*, 1738–1747.
- (154) Chung, S. J.; Hahn, T.; Klee, W. *Acta Crystallogr. A* **1984**, *40*, 42–50.
- (155) Shervashidze, N.; Schweitzer, P.; Van Leeuwen, E. J.; Mehlhorn, K.; Borgwardt, K. M. *J. Mach. Learn. Res.* **2011**, *12*.
- (156) Ongari, D.; Talirz, L.; Jablonka, K. M.; Siderius, D. W.; Smit, B. *J. Chem. Eng. Data* **2022**.
- (157) Bemis, G. W.; Murcko, M. A. *J. Med. Chem.* **1996**, *39*, 2887–2893.
- (158) Wilson, A. G.; Izmailov, P. *Adv. Neural Inf. Process Syst.* **2020**, *33*, 4697–4708.
- (159) Ward, L. et al. *Comput. Mater. Sci.* **2018**, *152*, 60–69.
- (160) Buitinck, L.; Louppe, G.; Blondel, M.; Pedregosa, F.; Mueller, A.; Grisel, O.; Niculae, V.; Prettenhofer, P.; Gramfort, A.; Grobler, J.; Layton, R.; Vanderplas, J.; Joly, A.; Holt, B.; Varoquaux, G. *arXiv preprint Arxiv-1309.0238* **2013**.
- (161) Prodan, E.; Kohn, W. *Proc. Natl. Acad. Sci. U.S.A.* **2005**, *102*, 11635–11638.
- (162) Poltavsky, I.; Tkatchenko, A. *J. Phys. Chem. Lett* **2021**, *12*, 6551–6564.
- (163) Janet, J. P.; Kulik, H. J. *J. Phys. Chem. A* **2017**, *121*, 8939–8954.
- (164) Bartók, A. P.; Kondor, R.; Csányi, G. *Phys. Rev. B* **2013**, *87*.
- (165) R. Zimmermann, N. E.; Jain, A. *RSC Adv.* **2020**, *10*, 6063–6081.
- (166) Widdowson, D.; Mosca, M. M.; Pulido, A.; Kurlin, V.; Cooper, A. I. *MATCH Commun. Math. Comput. Chem.* **2022**, *87*, 529–559.
- (167) Pettifor, D. *Solid State Commun.* **1984**, *51*, 31–34.
- (168) Glawe, H.; Sanna, A.; Gross, E. K. U.; Marques, M. A. L. *New J. Phys.* **2016**, *18*, 093011.
- (169) Hargreaves, C. J.; Dyer, M. S.; Gaultois, M. W.; Kurlin, V. A.; Rosseinsky, M. J. *Chem. Mater.* **2020**, *32*, 10610–10620.
- (170) Cumming, G., *Understanding the new statistics: Effect sizes, confidence intervals, and meta-analysis*; Routledge: 2013.
- (171) Landrum, G. et al. rdkit/rdkit: 2022_03_3 (Q1 2022) Release, available at <https://www.rdkit.org/> (accessed 2022-12-11), 2022.
- (172) Halder, P.; Prerna; Singh, J. K. *J. Chem. Inf. Model.* **2021**, *61*, 5827–5840.
- (173) Wicker, J. G. P.; Cooper, R. I. *J. Chem. Inf. Model.* **2016**, *56*, 2347–2352.
- (174) Ong, S. P.; Richards, W. D.; Jain, A.; Hautier, G.; Kocher, M.; Cholia, S.; Gunter, D.; Chevrier, V. L.; Persson, K. A.; Ceder, G. *Comput. Mater. Sci.* **2013**, *68*, 314–319.
- (175) Ongari, D.; Boyd, P. G.; Barthel, S.; Witman, M.; Haranczyk, M.; Smit, B. *Langmuir* **2017**, *33*, 14529–14538.

- (176) Willems, T. F.; Rycroft, C. H.; Kazi, M.; Meza, J. C.; Haranczyk, M. *Microporous Mesoporous Mater.* **2012**, *149*, 134–141.
- (177) Pinheiro, M.; Martin, R. L.; Rycroft, C. H.; Jones, A.; Iglesia, E.; Haranczyk, M. *J. Mol. Graph. Model.* **2013**, *44*, 208–219.
- (178) Jones, A. J.; Ostrouchov, C.; Haranczyk, M.; Iglesia, E. *Microporous Mesoporous Mater.* **2013**, *181*, 208–216.
- (179) Lee, Y.; Barthel, S. D.; Dłotko, P.; Moosavi, S. M.; Hess, K.; Smit, B. *Nat. Commun.* **2017**, *8*, 1–8.
- (180) Lee, Y.; Barthel, S. D.; Dłotko, P.; Moosavi, S. M.; Hess, K.; Smit, B. *J. Chem. Theory Comput.* **2018**, *14*, 4427–4437.
- (181) Krishnapriyan, A. S.; Haranczyk, M.; Morozov, D. *J. Phys. Chem. C* **2020**, *124*, 9360–9368.
- (182) Krishnapriyan, A. S.; Montoya, J.; Haranczyk, M.; Hummelshøj, J.; Morozov, D. *Sci. Rep.* **2021**, *11*, 8888.
- (183) Perea, J. A.; Munch, E.; Khasawneh, F. A. *Found. Comput. Math.* **2022**, 1–58.
- (184) Tymochko, S.; Munch, E.; Khasawneh, F. A. In *18th IEEE International Conference On Machine Learning And Applications, ICMLA 2019, Boca Raton, FL, USA, December 16–19, 2019*, ed. by Wani, M. A.; Khoshgoftaar, T. M.; Wang, D.; Wang, H.; Seliya, N., IEEE: 2019, pp 1227–1234.
- (185) Le, T. T.; Fu, W.; Moore, J. H. *Method. Biochem. Anal.* **2020**, *36*, 250–256.
- (186) Sawilowsky, S. S. *Journal of Modern Applied Statistical Methods* **2009**, *8*, 597–599.
- (187) Baird, S. G.; Jablonka, K. M.; Alverson, M. D.; Sayeed, H. M.; Khan, M. F.; Seegmiller, C.; Smit, B.; Sparks, T. D. *J. Open Source Softw.* **2022**, *7*, 4528.
- (188) Zeiler, M. D.; Fergus, R. In *Computer Vision – ECCV 2014*; Springer International Publishing: 2014, pp 818–833.
- (189) Hung, T.-H.; Xu, Z.-X.; Kang, D.-Y.; Lin, L.-C. *J. Phys. Chem. C* **2022**, *126*, 2813–2822.
- (190) Cho, E. H.; Lin, L.-C. *J. Phys. Chem. Lett* **2021**, *12*, 2279–2285.
- (191) Kim, B.; Lee, S.; Kim, J. *Sci. Adv.* **2020**, *6*, eaax9324.
- (192) Artrith, N.; Butler, K. T.; Coudert, F.-X.; Han, S.; Isayev, O.; Jain, A.; Walsh, A. *Nat. Chem.* **2021**, *13*, 505–508.
- (193) Wang, A. Y.-T.; Murdock, R. J.; Kauwe, S. K.; Oliynyk, A. O.; Gurlo, A.; Br-goch, J.; Persson, K. A.; Sparks, T. D. *Chem. Mater.* **2020**, *32*, 4954–4965.
- (194) Riley, P. *Nature* **2019**, *572*, 27–29.
- (195) Gropp, C.; Canossa, S.; Wuttke, S.; Gándara, F.; Li, Q.; Gagliardi, L.; Yaghi, O. M. *ACS Cent. Sci.* **2020**, *6*, 1255–1273.
- (196) Majumdar, S.; Moosavi, S. M.; Jablonka, K. M.; Ongari, D.; Smit, B. *ACS Appl. Mater. Interfaces* **2021**, *13*, 61004–61014.
- (197) Meredig, B.; Antono, E.; Church, C.; Hutchinson, M.; Ling, J.; Paradiso, S.; Blaiszik, B.; Foster, I.; Gibbons, B.; Hattrick-Simpers, J.; Mehta, A.; Ward, L. *Mol. Syst. Des. Eng.* **2018**, *3*, 819–825.
- (198) Xiong, Z.; Cui, Y.; Liu, Z.; Zhao, Y.; Hu, M.; Hu, J. *Comput. Mater. Sci.* **2020**, *171*, 109203.
- (199) Sheridan, R. P. *J. Chem. Inf. Model.* **2013**, *53*, 783–790.

- (200) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. *Chem. Sci.* **2018**, *9*, 513–530.
- (201) Kennard, R. W.; Stone, L. A. *Technometrics* **1969**, *11*, 137–148.
- (202) Ramsundar, B.; Eastman, P.; Walters, P.; Pande, V.; Leswing, K.; Wu, Z., *Deep Learning for the Life Sciences*; O'Reilly Media: 2019.
- (203) Pan, J.; Pham, V.; Dorairaj, M.; Chen, H.; Lee, J.-Y. In *arXiv preprint Arxiv-2004.03045*, 2020.
- (204) Banachewicz, K.; Massaron, L.; Goldbloom, A., *The Kaggle Book The Kaggle Book*; Packt Publishing: Birmingham, England, 2022.
- (205) Mitchell, M.; Wu, S.; Zaldivar, A.; Barnes, P.; Vasserman, L.; Hutchinson, B.; Spitzer, E.; Raji, I. D.; Gebru, T. In *Proceedings of the conference on fairness, accountability, and transparency*, 2019, pp 220–229.
- (206) Dyson, F. J. *Science* **2012**, *338*, 1426–1427.
- (207) Jablonka, K. M.; Jothiappan, G. M.; Wang, S.; Smit, B.; Yoo, B. *Nat. Commun.* **2021**, *12*.
- (208) Jablonka, K. M.; Patiny, L.; Smit, B. *Nat. Chem.* **2022**, *14*, 365–376.
- (209) Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A. V.; Gulin, A. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Curran Associates Inc.: Montréal, Canada, 2018, pp 6639–6649.
- (210) Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.
- (211) Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. *Adv. Neural Inf. Process Syst.* **2011**, *24*.
- (212) Dunn, A.; Wang, Q.; Ganose, A.; Dopp, D.; Jain, A. *npj Comput. Mater.* **2020**, *6*.
- (213) Olson, R. S.; Urbanowicz, R. J.; Andrews, P. C.; Lavender, N. A.; Kidd, L. C.; Moore, J. H. In Squillero, G., Burelli, P., Eds.; Springer International Publishing: 2016; Chapter Automating Biomedical Data Science Through Tree-Based Pipeline Optimization, pp 123–137.
- (214) Olson, R. S.; Bartley, N.; Urbanowicz, R. J.; Moore, J. H. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, ACM: Denver, Colorado, USA, 2016, pp 485–492.
- (215) Le, T. T.; Fu, W.; Moore, J. H. *Bioinformatics* **2019**, *36*, ed. by Kelso, J., 250–256.
- (216) Dubbeldam, D.; Calero, S.; Ellis, D. E.; Snurr, R. Q. *Mol. Simulat.* **2015**, *42*, 81–101.
- (217) Rappe, A. K.; Casewit, C. J.; Colwell, K. S.; Goddard, W. A.; Skiff, W. M. *J. Am. Chem. Soc.* **1992**, *114*, 10024–10035.
- (218) Jablonka, K. M.; Ongari, D.; Smit, B. *J. Chem. Theory Comput.* **2019**, *15*, 5635–5641.
- (219) Huber, S. P. et al. *Sci. Data* **2020**, *7*, 300.
- (220) Uhrin, M.; Huber, S. P.; Yu, J.; Marzari, N.; Pizzi, G. *Comput. Mater. Sci.* **2021**, *187*, 110086.
- (221) Talirz, L.; Kumbhar, S.; Passaro, E.; Yakutovich, A. V.; Granata, V.; Gargiulo, F.; Borelli, M.; Uhrin, M.; Huber, S. P.; Zoupanos, S., et al. *Sci. Data* **2020**, *7*, 1–12.

- (222) Jain, A.; Ong, S. P.; Hautier, G.; Chen, W.; Richards, W. D.; Dacek, S.; Cholia, S.; Gunter, D.; Skinner, D.; Ceder, G.; Persson, K. A. *APL Mater.* **2013**, *1*, 011002.
- (223) Huck, P.; Gunter, D.; Cholia, S.; Winston, D.; N'Diaye, A. T.; Persson, K. *Concurrency Computat.: Pract. Exper.* **2015**, *28*, 1982–1993.
- (224) Fernandez, M.; Trefiak, N. R.; Woo, T. K. *J. Phys. Chem. C* **2013**, *117*, 14095–14105.
- (225) Todeschini, R.; Consonni, V. In *Handbook of Chemoinformatics*; Wiley-VCH Verlag GmbH: Weinheim, Germany, 2008, pp 1004–1033.
- (226) Baumgärtner, A. *J. Chem. Phys.* **1993**, *98*, 7496–7501.
- (227) Jiang, Y.; Chen, D.; Chen, X.; Li, T.; Wei, G.-W.; Pan, F. *npj Comput. Mater.* **2021**, *7*, 28.
- (228) Wirth, M.; Volkamer, A.; Zoete, V.; Rippmann, F.; Michielin, O.; Rarey, M.; Sauer, W. H. B. *J. Comput. Aided Mol. Des.* **2013**, *27*, 511–524.
- (229) Arteca, G. A. In *Rev. Comput. Chem.* John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2007, pp 191–253.
- (230) Bucior, B. J.; Bobbitt, N. S.; Islamoglu, T.; Goswami, S.; Gopalan, A.; Yildirim, T.; Farha, O. K.; Bagheri, N.; Snurr, R. Q. *Mol. Syst. Des. Eng.* **2019**, *4*, 162–174.
- (231) Tosco, P.; Stiefl, N.; Landrum, G. *J. Cheminf.* **2014**, *6*, 1–4.
- (232) Sauer, W. H. B.; Schwarz, M. K. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 987–1003.
- (233) Wilmer, C. E.; Kim, K. C.; Snurr, R. Q. *J. Phys. Chem. Lett.* **2012**, *3*, 2506–2511.
- (234) Fabri, A.; Pion, S. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, 2009, pp 538–539.
- (235) Adams, H.; Emerson, T.; Kirby, M.; Neville, R.; Peterson, C.; Shipman, P.; Chepushtanova, S.; Hanson, E.; Motta, F.; Ziegelmeier, L. *J. Mach. Learn. Res.* **2017**, *18*, 1–35.
- (236) McInnes, L. scikit-tda/pervect: Vectorization of persistence diagrams and approximate Wasserstein distance, <https://github.com/scikit-tda/pervect>, accessed 2022-12-11, 2022.
- (237) Clancy, P. *ACS Cent. Sci.* **2020**, *6*, 464–466.
- (238) Manson, S. S., *Fatigue and durability of structural materials*; ASM International: Materials Park, Ohio, 2006.
- (239) Jablonka, K. M.; Ongari, D.; Moosavi, S. M.; Smit, B. *Chem. Rev.* **2020**, *120*, 8066–8129.
- (240) Kumar, J. N.; Li, Q.; Tang, K. Y.; Buonassisi, T.; Gonzalez-Oyarce, A. L.; Ye, J. *npj Comput. Mater.* **2019**, *5*, 1–6.
- (241) Chen, L.; Kim, C.; Batra, R.; Lightstone, J. P.; Wu, C.; Li, Z.; Deshmukh, A. A.; Wang, Y.; Tran, H. D.; Vashishta, P.; Sotzing, G. A.; Cao, Y.; Ramprasad, R. *npj Comput. Mater.* **2020**, *6*, 61.
- (242) Khadilkar, M. R.; Paradiso, S.; Delaney, K. T.; Fredrickson, G. H. *Macromolecules* **2017**, *50*, 6702–6709.
- (243) Wang, W.; Yang, T.; Harris, W. H.; Gómez-Bombarelli, R. *Chem. Commun.* **2020**, *56*, 8920–8923.

- (244) Settles, B. *Synth. Lect. Artif. Intell. Mach. Learn.* **2012**, 6, 1–114.
- (245) Janet, J. P.; Ramesh, S.; Duan, C.; Kulik, H. J. *ACS Cent. Sci.* **2020**, 6, 513–524.
- (246) Herbol, H. C.; Hu, W.; Frazier, P.; Clancy, P.; Poloczek, M. *npj Comput. Mater.* **2018**, 4, 51.
- (247) Häse, F.; Roch, L. M.; Kreisbeck, C.; Aspuru-Guzik, A. *ACS Cent. Sci.* **2018**, 4, 1134–1145.
- (248) Ju, S.; Shiga, T.; Feng, L.; Hou, Z.; Tsuda, K.; Shiomi, J. *Phys. Rev. X* **2017**, 7, 021024.
- (249) Griffiths, R.-R.; Hernández-Lobato, J. M. *Chem. Sci.* **2020**, 11, 577–586.
- (250) Pyzer-Knapp, E.; Day, G.; Chen, L.; Cooper, A. I. In *ChemRxiv preprint 10.26434/chemrxiv.13019960.v1*, 2020.
- (251) Frazier, P. I. In *arXiv preprint Arxiv-1807.02811*, 2018.
- (252) Zitzler, E.; Brockhoff, D.; Thiele, L. In *Evolutionary Multi-Criterion Optimization*, ed. by Obayashi, S.; Deb, K.; Poloni, C.; Hiroyasu, T.; Murata, T., Springer: Berlin, Heidelberg, 2007, pp 862–876.
- (253) Jackson, N. E.; Webb, M. A.; de Pablo, J. J. *Curr. Opin. Chem. Eng.* **2019**, 23, 106–114.
- (254) Ferguson, A. L. *J. Phys.: Condens. Matter* **2018**, 30, 043002.
- (255) Sherman, Z. M.; Howard, M. P.; Lindquist, B. A.; Jadrich, R. B.; Truskett, T. M. *J. Chem. Phys.* **2020**, 152, 140902.
- (256) Zuluaga, M.; Krause, A.; Püschel, M. *J. Mach. Learn. Res.* **2016**, 17, 1–32.
- (257) Zuluaga, M.; Sergeant, G.; Krause, A.; Püschel, M. In *Proceedings of the 30th International Conference on Machine Learning*, ed. by Dasgupta, S.; McAllester, D., PMLR: Atlanta, Georgia, USA, 2013; Vol. 28, pp 462–470.
- (258) In *Formulation of Disperse Systems*, Tadros, T. F., Ed.; Wiley-VCH Verlag GmbH & Co. KGaA: Weinheim, Germany, 2014, pp 45–54.
- (259) Israelachvili, J., *Intermolecular and Surface Forces*; Academic Press: Burlington, MA, 2011.
- (260) Webb, M. A.; Jackson, N. E.; Gil, P. S.; de Pablo, J. J. *Sci. Adv.* **2020**, 6, eabc6216.
- (261) Tadros, T., *Applied surfactants: principles and applications*; Wiley-VCH: Weinheim Germany, 2005.
- (262) Frenkel, D.; Smit, B. In *Understanding Molecular Simulation*; Elsevier: 2002, pp 167–200.
- (263) Dunstan, D. E. *Sci. Rep.* **2019**, 9, 1–9.
- (264) Larson, R., *The Structure and Rheology of Complex Fluids*; Oxford University Press: New York, 1999.
- (265) Upadhyay, R.; Murthy, N. S.; Hoop, C. L.; Kosuri, S.; Nanda, V.; Kohn, J.; Baum, J.; Gormley, A. J. *Macromolecules* **2019**, 52, 8295–8304.
- (266) Häse, F.; Roch, L. M.; Aspuru-Guzik, A. *TRECHEM* **2019**, 1, 282–291.
- (267) Bergstra, J.; Bengio, Y. *J. Mach. Learn. Res.* **2012**, 13, 25.
- (268) Alvarez, M. A.; Rosasco, L.; Lawrence, N. D., et al. *Found. Trends Mach. Learn.* **2012**, 4, 195–266.

- (269) Lundberg, S. M.; Lee, S.-I. In *Advances in Neural Information Processing Systems 30*, Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: 2017, pp 4765–4774.
- (270) Dill, K., *Molecular driving forces : statistical thermodynamics in biology, chemistry, physics, and nanoscience*; Garland Science: London New York, 2011.
- (271) Wagner, T.; Emmerich, M.; Deutz, A.; Ponweiser, W. In *Parallel Problem Solving from Nature, PPSN XI*, ed. by Schaefer, R.; Cotta, C.; Kolodziej, J.; Rudolph, G., Springer: Berlin, Heidelberg, 2010, pp 718–727.
- (272) Lee, A. pyDOE, <https://github.com/tisimst/pyDOE>, accessed 2022-12-11, 2020.
- (273) J. in 't Veld, P. EMC: Enhanced Monte Carlo, <http://montecarlo.sourceforge.net/emc/Welcome.html>, accessed 2022-12-11, 2020.
- (274) In't Veld, P. J.; Rutledge, G. C. *Macromolecules* **2003**, *36*, 7358–7365.
- (275) Plimpton, S. J. *Comput. Phys.* **1995**, *117*, 1–19.
- (276) Sidky, H.; Colón, Y. J.; Helfferich, J.; Sikora, B. J.; Bezik, C.; Chu, W.; Giberti, F.; Guo, A. Z.; Jiang, X.; Lequieu, J., et al. *J. Chem. Phys.* **2018**, *148*, 044104.
- (277) Izrailev, S.; Stepaniants, S.; Isralewitz, B.; Kosztin, D.; Lu, H.; Molnar, F.; Wriggers, W.; Schulten, K. In *Computational Molecular Dynamics: Challenges, Methods, Ideas*, Griebel, M., Keyes, D. E., Nieminen, R. M., Roose, D., Schlick, T., Deuffhard, P., Hermans, J., Leimkuhler, B., Mark, A. E., Reich, S., Skeel, R. D., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 1999; Vol. 4, pp 39–65.
- (278) Kumar, S.; Rosenberg, J. M.; Bouzida, D.; Swendsen, R. H.; Kollman, P. A. *J. Comput. Chem.* **1992**, *13*, 1011–1021.
- (279) Pedregosa, F. et al. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
- (280) GPy GPy: A Gaussian process framework in python, <http://github.com/SheffieldML/GPy>, accessed 2022-12-11, since 2012.
- (281) Vivek Nair epsilon-PAL, <https://github.com/FlashRepo/epsilon-PAL>, 2017.
- (282) Kluyver, T.; Ragan-Kelley, B.; Pérez, F.; Granger, B.; Bussonnier, M.; Frederic, J.; Kelley, K.; Hamrick, J.; Grout, J.; Corlay, S.; Ivanov, P.; Avila, D.; Abdalla, S.; Willing, C.; Team, J. D. In *IOS Press*, 2016, pp 87–90.
- (283) Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. **2017**, 3149–3157.
- (284) Hunter, J. D. *Comput. Sci. Eng.* **2007**, *9*, 90–95.
- (285) Novak, R.; Xiao, L.; Hron, J.; Lee, J.; Alemi, A. A.; Sohl-Dickstein, J.; Schoenholz, S. S. In *International Conference on Learning Representations*, 2020.
- (286) Rapin, J.; Teytaud, O. Nevergrad - A gradient-free optimization platform, <https://GitHub.com/FacebookResearch/Nevergrad>, accessed 2022-12-11, 2018.
- (287) Lam, S. K.; Pitrou, A.; Seibert, S. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC - LLVM '15*, ACM Press: Austin, Texas, 2015, pp 1–6.
- (288) Harris, C. R. et al. *Nature* **2020**, *585*, 357–362.
- (289) McKinney, W. In *Python in Science Conference*, Austin, Texas, 2010, pp 56–61.
- (290) Virtanen, P. et al. *Nat. Methods* **2020**, *17*, 261–272.

- (291) Solgi, R. Geneticalgorithm, <https://github.com/rmsolgi/geneticalgorithm>, accessed 2023-3-9, 2020.
- (292) Jensen, W. B. *J. Chem. Educ.* **2007**, *84*, 1418.
- (293) Wöhler, F. In *Grundriss Der Chemie: Unorganische Chemie*; Duncker und Humblot: Berlin, 1835, p 3.
- (294) Latimer, W. M., *The Oxidation States of the Elements and Their Potentials in Aqueous Solutions*, Second edition; Prentice-Hall Chemistry Series; Prentice-Hall: Englewood Cliffs, 1952.
- (295) *Nomenclature of Inorganic Chemistry. IUPAC Recommendations 2005*; Connelly, N. G., of Chemistry (Great Britain), R. S., of Pure, I. U., Chemistry, A., Eds.; Royal Society of Chemistry Publishing/IUPAC: Cambridge, UK, 2005.
- (296) Kroll, J. H.; Donahue, N. M.; Jimenez, J. L.; Kessler, S. H.; Canagaratna, M. R.; Wilson, K. R.; Altieri, K. E.; Mazzoleni, L. R.; Wozniak, A. S.; Bluhm, H.; Mysak, E. R.; Smith, J. D.; Kolb, C. E.; Worsnop, D. R. *Nat. Chem.* **2011**, *3*, 133–139.
- (297) Terrett, J. A.; Cuthbertson, J. D.; Shurtleff, V. W.; MacMillan, D. W. C. *Nature* **2015**, *524*, 330–334.
- (298) Jørgensen, C. K., *Oxidation Numbers and Oxidation States*; Springer Berlin Heidelberg: Berlin, Heidelberg, 1969.
- (299) Ball, P. *Nature* **2011**, *469*, 26–28.
- (300) *IUPAC Compendium of Chemical Terminology: Gold Book*, 2.1.0; Nič, M., Jirát, J., Košata, B., Jenkins, A., McNaught, A., Eds.; IUPAC: Research Triangle Park, NC, 2009.
- (301) Karen, P.; McArdle, P.; Takats, J. *Pure Appl. Chem.* **2016**, *88*, 831–839.
- (302) Brown, I. D. *Chem. Rev.* **2009**, *109*, 6858–6919.
- (303) Pauling, L. *J. Am. Chem. Soc.* **1947**, *69*, 542–553.
- (304) Shields, G. P.; Raithby, P. R.; Allen, F. H.; Motherwell, W. D. S. *Acta Crystallogr. B Struct. Sci.* **2000**, *56*, 455–465.
- (305) Reeves, M. G.; Wood, P. A.; Parsons, S. *Acta Crystallogr. B* **2019**, *75*, 1096–1105.
- (306) Taylor, R.; Wood, P. A. *Chem. Rev.* **2019**, *119*, 9427–9477.
- (307) O’Keeffe, M. *Acta Crystallogr. A Cryst. Phys. Diffr. Theor. Gen. Crystallogr.* **1979**, *35*, 772–775.
- (308) Walsh, A.; Sokol, A. A.; Buckeridge, J.; Scanlon, D. O.; Catlow, C. R. A. *J. Phys. Chem. Lett.* **2017**, *8*, 2074–2075.
- (309) Pan, H.; Ganose, A. M.; Horton, M.; Aykol, M.; Persson, K. A.; Zimmermann, N. E. R.; Jain, A. *Inorg. Chem.* **2021**, *60*, 1590–1603.
- (310) Conry, R. R. In *Encyclopedia of Inorganic Chemistry*; American Cancer Society: 2006.
- (311) Wang, L.; Maxisch, T.; Ceder, G. *Phys. Rev. B* **2006**, *73*, 195107.
- (312) Stevanović, V.; Lany, S.; Zhang, X.; Zunger, A. *Phys. Rev. B* **2012**, *85*, 115104.
- (313) Raebiger, H.; Lany, S.; Zunger, A. *Nature* **2008**, *453*, 763–766.
- (314) Bendix, J.; Brorson, M.; Schäffer, C. E. In *Coordination Chemistry*, Kauffman, G. B., Ed.; American Chemical Society: Washington, DC, 1994; Vol. 565, pp 213–225.
- (315) Jansen, M.; Wedig, U. *Angew. Chem. Int. Ed.* **2008**, *47*, 10026–10029.

- (316) Holgate, S. CSD Data Curation – The Human Touch - The Cambridge Crystallographic Data Centre (CCDC), <https://www.ccdc.cam.ac.uk/Community/blog/CSD-data-curation-the-human-touch/>, accessed 2023-3-9, 2019.
- (317) Allen, F. H.; Taylor, R. *Chem. Soc. Rev.* **2004**, 33, 463.
- (318) *Structure Correlation*; Bürgi, H.-B., Dunitz, J. D., Eds.; Wiley: 1994.
- (319) Janet, J. P.; Kulik, H. J. *J. Phys. Chem. A* **2017**, 121, 8939–8954.
- (320) Pauling, L. *J. Am. Chem. Soc.* **1929**, 51, 1010–1026.
- (321) Baur, W. H. *Trans. Am. Crystallogr. Assoc.* **1970**, 6, 129–155.
- (322) George, J.; Waroquiers, D.; Di Stefano, D.; Petretto, G.; Rignanese, G.-M.; Hautier, G. *Angew. Chem. Int. Ed.* **2020**, 59, 7569–7575.
- (323) Müller, P.; Köpke, S.; Sheldrick, G. M. *Acta Crystallogr. D Biol. Crystallogr.* **2003**, 59, 32–37.
- (324) Harvey, M. A.; Baggio, S.; Baggio, R. *Acta Cryst. Sect. A Found Cryst.* **2006**, 62, 1038–1042.
- (325) Brown, I. D. *J. Phys. Chem. A* **2011**, 115, 12638–12645.
- (326) Liu, S.; Grinberg, I.; Takenaka, H.; Rappe, A. M. *Phys. Rev. B* **2013**, 88, 104102.
- (327) Jahn, H.; Teller, E. *Proc. R. Soc. Lond. A* **1937**, 161, 220–235.
- (328) Gillespie, R. J.; Hargittai, I., *The VSEPR Model of Molecular Geometry*, Dover ed; Dover Publications: Mineola, N.Y, 2012.
- (329) Zimmermann, N. E. R.; Horton, M. K.; Jain, A.; Haranczyk, M. *Front. Mater.* **2017**, 4, 34.
- (330) Davies, D. W.; Butler, K. T.; Isayev, O.; Walsh, A. *Faraday Discuss.* **2018**, 211, 553–568.
- (331) Ward, L.; Liu, R.; Krishna, A.; Hegde, V. I.; Agrawal, A.; Choudhary, A.; Wolverton, C. *Phys. Rev. B* **2017**, 96.
- (332) Rokach, L. *Artif. Intell. Rev.* **2010**, 33, 1–39.
- (333) Ahmed, A.; Robertson, C. M.; Steiner, A.; Whittles, T.; Ho, A.; Dhanak, V.; Zhang, H. *RSC Adv.* **2016**, 6, 8902–8905.
- (334) Molnar, C., *Interpretable Machine Learning, A Guide for Making Black Box Models Explainable*, <https://christophm.github.io/interpretable-ml-book/>, 2019.
- (335) Barthelet, K.; Marrot, J.; Riou, D.; Férey, G. *Angew. Chem. Int. Ed.* **2002**, 41, 281–284.
- (336) Centrone, A.; Harada, T.; Speakman, S.; Hatton, T. A. *Small* **2010**, 6, 1598–1602.
- (337) Leclerc, H.; Devic, T.; Devautour-Vinot, S.; Bazin, P.; Audebrand, N.; Férey, G.; Daturi, M.; Vimont, A.; Clet, G. *J. Phys. Chem. C* **2011**, 115, 19828–19840.
- (338) Kozachuk, O.; Meilikhov, M.; Yushenko, K.; Schneemann, A.; Jee, B.; Kutatheyil, A. V.; Bertmer, M.; Sternemann, C.; Pöpl, A.; Fischer, R. A. *Eur. J. Inorg. Chem.* **2013**, 2013, 4546–4557.
- (339) Krakowiak, J.; Lundberg, D.; Persson, I. *Inorg. Chem.* **2012**, 51, 9598–9609.
- (340) Bloch, E. D.; Murray, L. J.; Queen, W. L.; Chavan, S.; Maximoff, S. N.; Bigi, J. P.; Krishna, R.; Peterson, V. K.; Grandjean, F.; Long, G. J.; Smit, B.; Bordiga, S.; Brown, C. M.; Long, J. R. *J. Am. Chem. Soc.* **2011**, 133, 14814–14822.
- (341) Janet, J. P.; Kulik, H. J. *Chem. Sci.* **2017**, 8, 5137–5152.

- (342) Jiang, L.; Levchenko, S. V.; Rappe, A. M. *Phys. Rev. Lett.* **2012**, *108*, 166403.
- (343) Sechidis, K.; Tsoumakas, G.; Vlahavas, I. In *Machine Learning and Knowledge Discovery in Databases*, Gunopulos, D., Hofmann, T., Malerba, D., Vazirgianis, M., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2011; Vol. 6913, pp 145–158.
- (344) Schreiber, J.; Bilmes, J.; Noble, W. S. *J. Mach. Learn. Res.* **2020**, *21*, 1–6.
- (345) Momma, K.; Izumi, F. *J. Appl. Cryst.* **2011**, *44*, 1272–1276.
- (346) Dubbeldam, D.; Calero, S.; Vlugt, T. J. H. *Mol. Simul.* **2018**, *44*, 653–676.
- (347) Jablonka, K. M.; Charalambous, C.; Fernandez, E. S.; Wiechers, G.; Monteiro, J.; Moser, P.; Smit, B.; Garcia, S. *Sci. Adv.* **2023**, *9*, eadc9576.
- (348) Reynolds, A. J.; Verheyen, T. V.; Adeloju, S. B.; Meuleman, E.; Feron, P. *Environ. Sci. Technol.* **2012**, *46*, 3643–3654.
- (349) Veltman, K.; Singh, B.; Hertwich, E. G. *Environ. Sci. Technol.* **2010**, *44*, 1496–1502.
- (350) Programme, I. G. G. R. ENVIRONMENTAL IMPACTS OF AMINE EMISSION DURING POST COMBUSTION CAPTURE, <https://www.globalccsinstitute.com/archive/hub/publications/106171/environmental-impacts-amine-emissions-post-combustion-capture.pdf>, accessed 2023-3-9, 2010.
- (351) Khakharia, P.; Mertens, J.; Abu-Zahra, M.; Vlugt, T.; Goetheer, E. In *Absorption-Based Post-combustion Capture of Carbon Dioxide*; Elsevier: 2016, pp 465–485.
- (352) Biegler, L., *Systematic methods of chemical process design*; Prentice Hall PTR: Upper Saddle River, N.J., 1997.
- (353) Ieaigh *Valuing Flexibility in CCS Power Plants*; tech. rep. 2017-09; Cheltenham, UK, 2017.
- (354) Flø, N. E.; Kvamsdal, H. M.; Hillestad, M. *Int. J. Greenh. Gas Con.* **2016**, *48*, 204–215.
- (355) Gaspar, J.; Jorgensen, J. B.; Fosbol, P. L. *IFAC-PapersOnLine* **2015**, *48*, 580–585.
- (356) Chalmers, H.; Leach, M.; Lucquiaud, M.; Gibbins, J. *Enrgy. Proced.* **2009**, *1*, 4289–4296.
- (357) Flø, N. E.; Kvamsdal, H. M.; Hillestad, M.; Mejdell, T. *Comput. Chem. Eng.* **2016**, *86*, 171–183.
- (358) Charalambous, C.; Saleh, A.; van der Spek, M.; Wiechers, G.; Moser, P.; Huizinga, A.; Gravesteijn, P.; Ros, J.; Monteiro, J. G. M.-S.; Goetheer, E.; Garcia, S. *SSRN Electronic Journal* **2021**.
- (359) Kachko, A.; van der Ham, L. V.; Geers, L. F. G.; Huizinga, A.; Rieder, A.; Abu-Zahra, M. R. M.; Vlugt, T. J. H.; Goetheer, E. L. V. *Ind. Eng. Chem. Res.* **2015**, *54*, 5769–5776.
- (360) Pearl, J., *Causality*; Cambridge University Press: 2009.
- (361) Moser, P.; Schmidt, S.; Sieder, G.; Garcia, H.; Stoffregen, T. *Int. J. Greenh. Gas Con.* **2011**, *5*, 620–627.
- (362) Moser, P.; Schmidt, S.; Stahl, K. In *Energy Procedia*, 2011; Vol. 4, pp 473–479.
- (363) Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Curran Associates Inc.: Long Beach, California, USA, 2017, pp 3149–3157.

- (364) Friedman, J. H. *Annals of statistics* **2001**, 1189–1232.
- (365) Das, K.; Krzywinski, M.; Altman, N. *Nat. Methods* **2019**, *16*, 451–452.
- (366) Bassett, G.; Koenker, R. *Econometrica* **1978**, *46*, 33–50.
- (367) Hernán, M. A.; Robins, J. M. *Causal inference*, 2010.
- (368) Brodersen, K. H.; Gallusser, F.; Koehler, J.; Remy, N.; Scott, S. L. *Ann. Appl. Stat.* **2015**, *9*, 247–274.
- (369) Hartono, A.; Svendsen, H. F.; Knuutila, H. K. *SSRN Journal* **2021**, 1–12.
- (370) Mertens, J.; Lepaumier, H.; Desagher, D.; Thielens, M.-L. *Int. J. Greenh. Gas Con.* **2013**, *13*, 72–77.
- (371) Khakharia, P.; Brachert, L.; Mertens, J.; Anderlohr, C.; Huizinga, A.; Fernandez, E. S.; Schallert, B.; Schaber, K.; Vlugt, T. J.; Goetheer, E. *Int. J. Greenh. Gas Con.* **2015**, *34*, 63–74.
- (372) Ciftja, A. F.; Hartono, A.; da Silva, E. F.; Svendsen, H. F. *Enrgy. Proced.* **2011**, *4*, 614–620.
- (373) Lookman, T.; Balachandran, P. V.; Xue, D.; Yuan, R. *npj Comput. Mater.* **2019**, *5*, 21.
- (374) Schweidtmann, A. M.; Esche, E.; Fischer, A.; Kloft, M.; Repke, J.-U.; Sager, S.; Mitsos, A. *Chem-ing-tech.* **2021**, *93*, 2029–2039.
- (375) Weber, J. M.; Guo, Z.; Zhang, C.; Schweidtmann, A. M.; Lapkin, A. A. *Chem. Soc. Rev.* **2021**, *50*, 12013–12036.
- (376) Moser, P.; Schmidt, S.; Stahl, K.; Vorberg, G.; Lozano, G. A.; Stoffregen, T.; Rösler, F. *Enrgy. Proced.* **2014**, *63*, 902–910.
- (377) Rieder, A.; Dhingra, S.; Khakharia, P.; Zangrilli, L.; Schallert, B.; Irons, R.; Unterberger, S.; Van Os, P.; Goetheer, E. In *Enrgy. Proced.* 2017; Vol. 114, pp 1195–1209.
- (378) Da Silva, E. F.; Hoff, K. A.; Booth, A. *Enrgy. Proced.* **2013**, *37*, 784–790.
- (379) Bui, M.; Gunawan, I.; Verheyen, V.; Feron, P.; Meuleman, E.; Adeloju, S. *Comput. Chem. Eng.* **2014**, *61*, 245–265.
- (380) Bui, M.; Flø, N. E.; de Cazenove, T.; Dowell, N. M. *Int. J. Greenh. Gas Con.* **2020**, *93*, 102879.
- (381) Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. *Adv. Neural Inf. Process Syst.* **2017**, *30*, 3146–3154.
- (382) Herzen, J. et al. *J. Mach. Learn. Res.* **2022**, *23*, 1–6.
- (383) Van Rossum, G.; Drake, F. L., *Python 3 Reference Manual*; CreateSpace: Scotts Valley, CA, 2009.
- (384) Seabold, S.; Perktold, J. In *9th Python in Science Conference*, 2010.
- (385) Harris, C. R. et al. *Nature* **2020**, *585*, 357–362.
- (386) Paszke, A. et al. In *Advances in Neural Information Processing Systems 32*, Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: 2019, pp 8024–8035.
- (387) Lundberg, S. M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J. M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.-I. *Nat. Mach. Intell.* **2020**, *2*, 2522–5839.
- (388) Jablonka, K. M.; Schwaller, P.; Ortega-Guerrero, A.; Smit, B. In *ChemRxiv preprint 10.26434/chemrxiv-2023-fw8n4*, 2023.

- (389) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
- (390) Chowdhery, A. et al. In *arXiv preprint Arxiv-2204.02311*, 2022.
- (391) Hoffmann, J. et al. In *arXiv preprint Arxiv-2203.15556*, 2022.
- (392) Edwards, C. N.; Lai, T.; Ros, K.; Honke, G.; Ji, H. *Conference On Empirical Methods In Natural Language Processing* **2022**.
- (393) Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Nee-lakantan, A.; Shyam, P.; Sastry, G.; Askell, A., et al. *Adv. Neural Inf. Process Syst.* **2020**, *33*, 1877–1901.
- (394) Hocky, G. M.; White, A. D. *Digital Discovery* **2022**, *1*, 79–83.
- (395) White, A. D.; Hocky, G. M.; Gandhi, H. A.; Ansari, M.; Cox, S.; Wellawatte, G. P.; Sasmal, S.; Yang, Z.; Liu, K.; Singh, Y.; Ccoa, W. J. P. *Digital Discovery* **2023**.
- (396) Taylor, R.; Kardas, M.; Cucurull, G.; Scialom, T.; Hartshorn, A.; Saravia, E.; Poulton, A.; Kerkez, V.; Stojnic, R. In *arXiv Arxiv-2211.09085*, 2022.
- (397) Dunn, A.; Dagdelen, J.; Walker, N.; Lee, S.; Rosen, A. S.; Ceder, G.; Persson, K.; Jain, A. In *arXiv preprint Arxiv-2212.05238*, 2022.
- (398) Dinh, T.; Zeng, Y.; Zhang, R.; Lin, Z.; Gira, M.; Rajput, S.; Sohn, J.-y.; Papail-iopoulos, D.; Lee, K. In *arXiv preprint Arxiv-2206.06565*, 2022.
- (399) Howard, J.; Ruder, S. In *Proceedings of the 56th Annual Meeting of the As-sociation for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics: Melbourne, Australia, 2018, pp 328–339.
- (400) Pei, Z.; Yin, J.; Hawk, J. A.; Alman, D. E.; Gao, M. C. *npj Comput. Mater.* **2020**, *6*, 50.
- (401) Wang, A. Y.-T.; Kauwe, S. K.; Murdock, R. J.; Sparks, T. D. *npj Comput. Mater.* **2021**, *7*, 77.
- (402) Wang, Y.; Wang, J.; Cao, Z.; Farimani, A. B. *Nat. Mach. Intell.* **2022**, *4*, 279–287.
- (403) Breuck, P.-P. D.; Evans, M. L.; Rignanese, G.-M. *J. Phys.: Condens. Matter* **2021**, *33*, 404002.
- (404) Hollmann, N.; Müller, S.; Eggensperger, K.; Hutter, F. In *arXiv preprint Arxiv-2207.01848*, 2022.
- (405) Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P. J. *J. Mach. Learn. Res.* **2020**, *21*, 5485–5551.
- (406) Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. In 2019.
- (407) Mobley, D. L.; Guthrie, J. P. *J. Comput. Aided Mol. Des.* **2014**, *28*, 711–720.
- (408) Delaney, J. S. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1000–1005.
- (409) Mitchell, J. B. O. DLS-100 Solubility Dataset, [https://risweb.st-andrews.ac.uk:443-/portal/en/datasets/dls100-solubility-dataset\(3a3a5abc-8458-4924-8e6c-b804347605e8\).html](https://risweb.st-andrews.ac.uk:443-/portal/en/datasets/dls100-solubility-dataset(3a3a5abc-8458-4924-8e6c-b804347605e8).html), 2017.
- (410) Walters, P. Predicting Aqueous Solubility - It's Harder Than It Looks, <https://practicalcheminformatics.blogspot.com/2018/09/predicting-aqueous-solubility-its.html>, accessed 2023-02-06, Predicting Aqueous Solubility - It's Harder Than It Looks, 2018.
- (411) Bento, A. P.; Gaulton, A.; Hersey, A.; Bellis, L. J.; Chambers, J.; Davies, M.; Krüger, F. A.; Light, Y.; Mak, L.; McGlinchey, S., et al. *Nucleic Acids Res.* **2014**, *42*, D1083–D1090.

- (412) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B., et al. *Nucleic Acids Res.* **2012**, *40*, D1100–D1107.
- (413) Isert, C.; Atz, K.; Jiménez-Luna, J.; Schneider, G. QMugs: Quantum Mechanical Properties of Drug-like Molecules, <https://www.research-collection.ethz.ch/handle/20.500.11850/482129>, 2021.
- (414) Isert, C.; Atz, K.; Jiménez-Luna, J.; Schneider, G. *Sci. Data* **2022**, *9*, 273.
- (415) Nagasawa, S.; Al-Naamani, E.; Saeki, A. *J. Phys. Chem. Lett.* **2018**, *9*, 2639–2646.
- (416) *Nonequilibrium phase diagrams of ternary amorphous alloys*; Kawazoe, Y., Yu, J.-Z., Tsai, A.-P., Masumoto, T., Eds.; Landolt-Börnstein: Numerical Data and Functional Relationships in Science and Technology - New Series; Springer: New York, NY, 2006.
- (417) Zhuo, Y.; Tehrani, A. M.; Brgoch, J. *J. Phys. Chem. Lett.* **2018**, *9*, 1668–1673.
- (418) Ahneman, D. T.; Estrada, J. G.; Lin, S.; Dreher, S. D.; Doyle, A. G. *Science* **2018**, *360*, 186–190.
- (419) Perera, D.; Tucker, J. W.; Brahmabhatt, S.; Helal, C. J.; Chong, A.; Farrell, W.; Richardson, P.; Sach, N. W. *Science* **2018**, *359*, 429–434.
- (420) Schwaller, P.; Laino, T.; Gaudin, T.; Bolgar, P.; Hunter, C. A.; Bekas, C.; Lee, A. A. *ACS Cent. Sci.* **2019**, *5*, 1572–1583.
- (421) Winter, B.; Winter, C.; Schilling, J.; Bardow, A. *Digital Discovery* **2022**, *1*, 859–869.
- (422) Dai, D.; Sun, Y.; Dong, L.; Hao, Y.; Sui, Z.; Wei, F. In *arXiv preprint Arxiv-2212.10559*, 2022.
- (423) Weininger, D. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36.
- (424) Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Aspuru-Guzik, A. *Mach. Learn.: Sci. Technol.* **2020**, *1*, 045024.
- (425) Krenn, M. et al. *Patterns* **2022**, *3*, 100588.
- (426) Sanchez-Lengeling, B.; Aspuru-Guzik, A. *Science* **2018**, *361*, 360–365.
- (427) Yao, Z.; Sánchez-Lengeling, B.; Bobbitt, N. S.; Bucior, B. J.; Kumar, S. G. H.; Collins, S. P.; Burns, T.; Woo, T. K.; Farha, O. K.; Snurr, R. Q.; Aspuru-Guzik, A. *Nat. Mach. Intell.* **2021**, *3*, 76–86.
- (428) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. *ACS Cent. Sci.* **2018**, *4*, 268–276.
- (429) Lee, S.; Kim, B.; Kim, J. *J. Mater. Chem. A Mater. Energy Sustain.* **2019**, *7*, 2709–2716.
- (430) Nigam, A.; Friederich, P.; Krenn, M.; Aspuru-Guzik, A. *ICLR* **2019**.
- (431) Jablonka, K. M.; Mcilwaine, F.; Garcia, S.; Smit, B.; Yoo, B. In *arXiv preprint Arxiv-2102.00700*, 2021.
- (432) Chung, Y. G.; Gómez-Gualdrón, D. A.; Li, P.; Leperi, K. T.; Deria, P.; Zhang, H.; Vermeulen, N. A.; Stoddart, J. F.; You, F.; Hupp, J. T.; Farha, O. K.; Snurr, R. Q. *Sci. Adv.* **2016**, *2*, e1600909.
- (433) Collins, S. P.; Daff, T. D.; Piotrkowski, S. S.; Woo, T. K. *Sci. Adv.* **2016**, *2*, e1600954.
- (434) Ertl, P.; Rohde, B. *J. Cheminform.* **2012**, *4*, 12.

- (435) Preuer, K.; Renz, P.; Unterthiner, T.; Hochreiter, S.; Klambauer, G. *J. Chem. Inf. Model.* **2018**, *58*, 1736–1741.
- (436) Ertl, P.; Schuffenhauer, A. *J. Cheminform.* **2009**, *1*, 8.
- (437) Probst, D.; Reymond, J.-L. *J. Cheminform.* **2020**, *12*, 12.
- (438) Probst, D.; Reymond, J.-L. *J. Cheminform.* **2018**, *10*, 66.
- (439) Bannwarth, C.; Ehlert, S.; Grimme, S. *J. Chem. Theory Comput.* **2019**, *15*, 1652–1671.
- (440) Westermayr, J.; Gilkes, J.; Barrett, R.; Maurer, R. *J. Nat. Comput. Sci.* **2023**.
- (441) Krenn, M.; Pollice, R.; Guo, S. Y.; Aldeghi, M.; Cervera-Lierta, A.; Friederich, P.; dos Passos Gomes, G.; Häse, F.; Jinich, A.; Nigam, A.; Yao, Z.; Aspuru-Guzik, A. *Nat. Rev. Phys.* **2022**, *4*, 761–769.
- (442) Shekar, V.; Nicholas, G.; Najeeb, M. A.; Zeile, M.; Yu, V.; Wang, X.; Slack, D.; Li, Z.; Nega, P. W.; Chan, E. M.; Norquist, A. J.; Schrier, J.; Friedler, S. A. *J. Chem. Phys.* **2022**, *156*, 064108.
- (443) Sherck, N.; Shen, K.; Nguyen, M.; Yoo, B.; Köhler, S.; Speros, J. C.; Delaney, K. T.; Shell, M. S.; Fredrickson, G. H. *ACS Macro Lett.* **2021**, *10*, 576–583.
- (444) Ying, Z.; You, J.; Morris, C.; Ren, X.; Hamilton, W.; Leskovec, J. In *Advances in neural information processing systems*, 2018; Vol. 31.
- (445) Gabriel, R. *Lisp: Good News, Bad News, How to Win Big* **1991**, *2*.
- (446) Stonebraker, M. Tamr White Paper – The Seven Tenets of Scalable Data Unification, <http://www.tamr.com/wp-content/uploads/2017/06/The-Seven-Tenets-of-Scalable-Data-Unification-WP.pdf>, 2019.
- (447) Raymond, E., *The cathedral & the bazaar: musings on Linux and open source by an accidental revolutionary*; O'Reilly: Beijing, 1999.
- (448) Turner, J. Open Source Has a Funding Problem, <https://stackoverflow.blog/2021/01/07/open-source-has-a-funding-problem/>, accessed 2023-3-9, 2021.
- (449) Stannard, A. How to Build Sustainable Open Source Software Projects, <http://www.aaronstannard.com/sustainable-open-source-software/>, accessed 2023-3-9, 2020.
- (450) Gowers, T.; Nielsen, M. *Nature* **2009**, *461*, 879–881.
- (451) Epstein, D., *Range: why generalists triumph in a specialized world*; Riverhead Books: New York, 2019.
- (452) Gowers, T. Questions of Procedure, <https://gowers.wordpress.com/2009/02/01/questions-of-procedure/>, accessed 2023-3-9, 2009.
- (453) Kaggle Inc Kaggle: Your Home for Data Science, <https://www.kaggle.com/>, 2021.
- (454) Hanwell, M. D.; de Jong, W. A.; Harris, C. J. *J. Cheminf.* **2017**, *9*.
- (455) Brooks, B. J.; Thorn, A. L.; Smith, M.; Matthews, P.; Chen, S.; O'Steen, B.; Adams, S. E.; Townsend, J. A.; Murray-Rust, P. *J. Cheminf.* **2011**, *3*, 45.
- (456) Knight, N. J.; Kanza, S.; Cruickshank, D.; Brocklesby, W. S.; Frey, J. G. *IEEE IoT* **2020**, *7*, 8631–8640.
- (457) Schäfer, B. A.; Poetz, D.; Kramer, G. W. *JALA* **2004**, *9*, 375–381.
- (458) Schäfer, B. *Wiley Analytical Science* **2018**.
- (459) Miles, B.; Lee, P. L. *SLAS TECHNOLOGY: Translating Life Sciences Innovation* **2018**, *23*, 432–439.
- (460) Michel, K.; Meredig, B. *MRS Bull.* **2016**, *41*, 617–623.

- (461) Frenkel, M.; Diky, V.; Chirico, R. D.; Goldberg, R. N.; Heerklotz, H.; Ladbury, J. E.; Remeta, D. P.; Dymond, J. H.; Goodwin, A. R.; Marsh, K. N.; Wakeham, W. A.; Stein, S. E.; Brown, P. L.; Koenigsberger, E.; Williams, P. A. *J. Chem. Eng. Data* **2011**, *56*, 307–316.
- (462) Murray-Rust, P.; Rzepa, H. S. *J. Cheminf.* **2011**, *3*.
- (463) Murray-Rust, P.; Rzepa, H. S.; Wright, M. *New J. Chem.* **2001**, *25*, 618–634.
- (464) Kuhn, S.; Helmus, T.; Lancashire, R. J.; Murray-Rust, P.; Rzepa, H. S.; Steinbeck, C.; Willighagen, E. L. *J. Chem. Inf. Model.* **2007**, *47*, 2015–2034.
- (465) Anderson, A.; McGibbon, G.; Paramonov, A.; Bhal, S. *Looking Beyond Analytical Data Standardization - the Fourth Paradigm*; tech. rep., accessed 2023-3-9; ACD Labs, p 8.
- (466) Davies, A. N.; Lancashire, R. *Spectrosc. Eur.* **2017**, *29*, 3.
- (467) Bruno, I.; Frey, J. G. *Chem. Int.* **2017**, *39*, 5–8.
- (468) Aitsi-Selmi, A.; Blanchard, K.; Murray, V. *Palgrave Commun.* **2016**, *2*, 16016.
- (469) Andersen, C. et al. The OPTIMADE Specification, version 1.0, 2020.
- (470) McEwen, L. R. *Chem. Int.* **2020**, *42*, 15–17.
- (471) Grasselli, J. G. Jcamp-Dx, a Standard Format for Exchange of Infrared Spectra in Computer Readable Form, 2016.
- (472) Matthews, L.; Miller, T. *JALA: Journal of the Association for Laboratory Automation* **2000**, *5*, 60–61.
- (473) Ulrich, E. L.; Argentar, D.; Klimowicz, A.; Westler, W. M.; Markley, J. L. *Acta Crystallogr. A* **1996**, *52*, C577–C577.
- (474) Varde, A. S.; Begley, E. F.; Fahrenholz-Mann, S. In *Proceedings of the 4th International Workshop on Data Mining Standards, Services and Platforms*, Association for Computing Machinery: Philadelphia, Pennsylvania, 2006, pp 47–54.
- (475) Rühl, M. A.; Schaefer, R.; Kramer, G. W. *JALA* **2001**, *6*, 76–82.
- (476) Frenkel, M.; Chirico, R. D.; Diky, V. V.; Dong, Q.; Frenkel, S.; Franchois, P. R.; Embry, D. L.; Teague, T. L.; Marsh, K. N.; Wilhoit, R. C. *J. Chem. Eng. Data* **2003**, *48*, 2–13.
- (477) Pedrioli, P. G. A. et al. *Nat. Biotechnol.* **2004**, *22*, 1459–1466.
- (478) Orchard, S.; Hermjakob, H.; Taylor, C.; Binz, P.-A.; Hoogland, C.; Julian, R.; Garavelli, J. S.; Aebersold, R.; Apweiler, R. *PROTEOMICS* **2006**, *6*, 738–741.
- (479) Deutsch, E. *PROTEOMICS* **2008**, *8*, 2776–2777.
- (480) Wilhelm, M.; Kirchner, M.; Steen, J. A.; Steen, H. *Mol. Cell. Proteomics* **2012**, *11*, O111.011379.
- (481) Bouyssié, D.; Dubois, M.; Nasso, S.; de Peredo, A. G.; Burlet-Schiltz, O.; Aebersold, R.; Monsarrat, B. *Mol. Cell. Proteomics* **2015**, *14*, 771–781.
- (482) Bates, M.; Berliner, A. J.; Lachoff, J.; Jaschke, P. R.; Groban, E. S. *ACS Synth. Biol.* **2016**, *6*, 167–171.
- (483) Pupier, M. et al. *Magn. Reson. Chem.* **2018**, *56*, 703–715.
- (484) Schober, D. et al. *Anal. Chem.* **2017**, *90*, 649–656.
- (485) Bhamber, R. S.; Jankevics, A.; Deutsch, E. W.; Jones, A. R.; Dowsey, A. W. *J. Proteome Res.* **2020**, *20*, 172–183.
- (486) Rose, M. E.; Kitchin, J. R. *SoftwareX* **2019**, *10*, 100263.

- (487) Manz, T. A.; Limas, N. G. *RSC Adv.* **2016**, *6*, 47771–47801.
- (488) Limas, N. G.; Manz, T. A. *RSC Adv.* **2016**, *6*, 45727–45747.
- (489) Manz, T. A. *RSC Adv.* **2017**, *7*, 45552–45581.
- (490) Limas, N. G.; Manz, T. A. *RSC Adv.* **2018**, *8*, 2678–2707.
- (491) Potoff, J. J.; Siepmann, J. I. *Aiche J.* **2001**, *47*, 1676–1682.
- (492) Michels, A.; de Graaff, W.; Ten Seldam, C. A. *Physica* **1960**, *26*, 393–408.
- (493) Darkrim, F.; Levesque, D. *J. Chem. Phys.* **1998**, *109*, 4981–4984.
- (494) Martin, M. G.; Siepmann, J. I. *J. Phys. Chem. B* **1998**, *102*, 2569–2577.
- (495) Moghadam, P. Z.; Islamoglu, T.; Goswami, S.; Exley, J.; Fantham, M.; Kamin-ski, C. F.; Snurr, R. Q.; Farha, O. K.; Fairen-Jimenez, D. *Nat. Commun.* **2018**, *9*, 1378.
- (496) Zhang, L.; Siepmann, J. I. *Theor. Chem. Acc.* **2006**, *115*, 391–397.
- (497) Boato, G.; Casanova, G. *Physica* **1961**, *27*, 571–589.
- (498) Abascal, J. L.; Vega, C. *J. Chem. Phys.* **2005**, *123*, 234505.
- (499) Cho, E. H.; Lin, L.-C. *J. Chem. Theory Comput.* **2019**, *15*, 6323–6332.
- (500) Hedges, L. V. *J. Educ. Stat.* **1981**, *6*, 107–128.
- (501) Delgado-Friedrichs, O.; O’Keeffe, M. *Acta Cryst Sect A* **2003**, *59*, 351–360.
- (502) Del Rosario, Z.; Rupp, M.; Kim, Y.; Antono, E.; Ling, J. *J. Chem. Phys.* **2020**, *153*, 024112.
- (503) Moffaert, K. V.; Nowé, A. *J. Mach. Learn. Res.* **2014**, *15*, 3663–3692.
- (504) Forrester, A. I. J.; Söbester, A.; Keane, A. J., *Engineering Design via Surrogate Modelling: A Practical Guide*, First; Wiley: 2008.
- (505) Keane, A. J. *AIAA J.* **2006**, *44*, 879–891.
- (506) Ishibuchi, H.; Imada, R.; Setoguchi, Y.; Nojima, Y. *Evol. Comput.* **2018**, *26*, 411–440.
- (507) Frazier, P. I. In *Recent Advances in Optimization and Modeling of Contemporary Problems*, Gel, E., Ntamo, L., Shier, D., Greenberg, H. J., Eds.; INFORMS: 2018, pp 255–278.
- (508) Wu, J.; Frazier, P. In *Advances in Neural Information Processing Systems 32*, Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: 2019, pp 9813–9823.
- (509) Rekvig, L.; Kranenburg, M.; Vreede, J.; Hafskjold, B.; Smit, B. *Langmuir* **2003**, *19*, 8195–8205.
- (510) Goicochea, A. G. *Langmuir* **2007**, *23*, 11656–11663.
- (511) Español, P.; Warren, P. *Europhys. Lett.* **1995**, *30*, 191–196.
- (512) McInnes, L.; Healy, J.; Melville, J. In *arXiv preprint Arxiv-1802.03426*, 2018.
- (513) Karvonen, T.; Wynne, G.; Tronarp, F.; Oates, C. J.; Särkkä, S. In *arXiv preprint Arxiv-2001.10965*, 2020.
- (514) Roy, N.; McCallum, A. In *Proceedings of the Eighteenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2001, pp 441–448.
- (515) Cohn, D. A.; Ghahramani, Z.; Jordan, M. I. In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, MIT Press: Denver, Colorado, 1994, pp 705–712.

- (516) Lewis, D. D.; Gale, W. A. In *SIGIR '94*; Springer London: 1994, pp 3–12.
- (517) Gal, Y.; Ghahramani, Z. In *Proceedings of The 33rd International Conference on Machine Learning*, ed. by Balcan, M. F.; Weinberger, K. Q., PMLR: New York, New York, USA, 2016; Vol. 48, pp 1050–1059.
- (518) Ribeiro, M. T.; Singh, S.; Guestrin, C. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13–17, 2016*, ed. by Krishnapuram, B.; Shah, M.; Smola, A. J.; Aggarwal, C. C.; Shen, D.; Rastogi, R., ACM: 2016, pp 1135–1144.
- (519) Lee, A. Pyswarm, <https://github.com/tisimst/pyswarm>, accessed 2022-12-11, 2020.
- (520) Knuth, D., *The art of computer programming*; Addison-Wesley: Reading (Mass.) Menlo Park (Calif.) London etc, 1968.
- (521) Goldsmith, J.; Wong-Foy, A. G.; Cafarella, M. J.; Siegel, D. J. *Chem. Mater.* **2013**, *25*, 3373–3382.
- (522) Batten, S. R.; Champness, N. R.; Chen, X.-M.; Garcia-Martinez, J.; Kitagawa, S.; Öhrström, L.; O’Keeffe, M.; Suh, M. P.; Reedijk, J. *CrystEngComm* **2012**, *14*, 3001.
- (523) Chen, T.; Manz, T. A. *RSC Adv.* **2020**, *10*, 26944–26951.
- (524) Sturluson, A.; Huynh, M. T.; Kaija, A. R.; Laird, C.; Yoon, S.; Hou, F.; Feng, Z.; Wilmer, C. E.; Colón, Y. J.; Chung, Y. G.; Siderius, D. W.; Simon, C. M. *Mol. Simulat.* **2019**, *45*, 1082–1121.
- (525) Robin, M. B.; Day, P. In *Advances in Inorganic Chemistry and Radiochemistry*; Elsevier: 1968; Vol. 10, pp 247–422.
- (526) Yu, H. S.; Truhlar, D. G. *Angew. Chem. Int. Ed.* **2016**, *55*, 9004–9006.
- (527) Zeng, J.-Y.; Wang, X.-S.; Qi, Y.-D.; Yu, Y.; Zeng, X.; Zhang, X.-Z. *Angew. Chem. Int. Ed.* **2019**, *58*, 5692–5696.
- (528) Uvarova, M. A.; Grineva, A. A.; Datchuk, R. R.; Nefedov, S. E. *Russ. J. Inorg. Chem.* **2018**, *63*, 618–625.
- (529) Baenziger, N. C.; Struss, A. W. *Inorg. Chem.* **1976**, *15*, 1807–1809.
- (530) Hu, J.; Zhang, J.; Zhao, J.; Hu, L.; Chen, S. *J. Coord. Chem.* **2016**, *69*, 574–584.
- (531) Zhu, H.-F.; Fan, J.; Okamura, T.-a.; Sun, W.-Y.; Ueyama, N. *Cryst. Growth Des.* **2005**, *5*, 289–294.
- (532) Bai, Y.; Wang, J.-L.; Dang, D.-B.; Li, M.-M.; Niu, J.-Y. *CrystEngComm* **2012**, *14*, 1575–1581.
- (533) Nasser, N.; Puddephatt, R. J. *Chem. Commun.* **2011**, *47*, 2808.
- (534) You, Z.-L.; Zhu, H.-L.; Liu, W.-S. *Acta Crystallogr. C Cryst. Struct. Commun* **2004**, *60*, m620–m622.
- (535) Zhao, J.; Hu, J.; Bai, Y.; Chen, S.; Li, S. *J. Coord. Chem.* **2012**, *65*, 3216–3226.
- (536) Shin, J. W.; Lee, Y. H.; Harrowfield, J.; Hayami, S.; Kim, Y. *Polyhedron* **2017**, *130*, 94–99.
- (537) Zhang, L.-H.; Liu, Y.-Y.; Ma, J.-F.; Yang, J.; Zhang, L.-G.; Li, J.; Li, Y.-W. *Polyhedron* **2011**, *30*, 764–777.
- (538) Wang, H.; Wan, C.-Q.; Yang, J.; Mak, T. C. W. *Cryst. Growth Des.* **2014**, *14*, 3530–3540.

- (539) Fang, Q.; Zhu, G.; Xue, M.; Wang, Z.; Sun, J.; Qiu, S. *Cryst. Growth Des.* **2008**, *8*, 319–329.
- (540) Lu, L.; Wang, J.; Xie, B.; Liu, J.-Q.; Yadav, R.; Singh, A.; Kumar, A. *New J. Chem.* **2017**, *41*, 3537–3542.
- (541) Fan, Z.-W.; Li, L.; Cui, K.; Yang, S.-S.; Han, F.-Q. *Synth. React. Inorg., Met.-Org., Nano-Met. Chem.* **2016**, *46*, 1701–1704.
- (542) Gu, J.-Z.; Liang, X.-X.; Cui, Y.-H.; Wu, J.; Shi, Z.-F.; Kirillov, A. M. *CrystEngComm* **2017**, *19*, 2570–2588.
- (543) Giniyatullina, Y. R.; Peresyphkina, E. V.; Virovets, A. V.; Cherkasova, T. G.; Tatarinova, E. S. *Russ. J. Inorg. Chem.* **2012**, *57*, 811–814.
- (544) Bai, Z.; Wang, Y.; Liu, W.; Li, Y.; Xie, J.; Chen, L.; Sheng, D.; Diwu, J.; Chai, Z.; Wang, S. *Cryst. Growth Des.* **2017**, *17*, 3847–3853.
- (545) Yang, X.-J.; Bao, S.-S.; Zheng, T.; Zheng, L.-M. *Chem. Commun.* **2012**, *48*, 6565.
- (546) Truong, K.-N.; Merckens, C.; Englert, U. *Acta Crystallogr. C Struct. Chem.* **2017**, *73*, 724–730.
- (547) Zhou, H.; Chen, Q.; Yuan, A.-H.; Zhou, H.-B.; Shen, X.-P.; Chen, L.; Song, Y. *Cryst. Growth Des.* **2017**, *17*, 6523–6530.
- (548) Zhang, M.-B.; Chen, Z.-L.; Hu, R.-X.; Liang, F.-P.; Zhou, Z.-Y. *Chin. J. Chem.* **2006**, *24*, 193–198.
- (549) Hawthorne, F. C.; Borys, I.; Ferguson, R. B. *Acta Crystallogr. C Cryst. Struct. Commun.* **1983**, *39*, 540–542.
- (550) Lama, P.; Sañudo, E. C.; Bharadwaj, P. K. *Dalton Trans.* **2012**, *41*, 2979.
- (551) Mak, T.; Yip, W.; Kennard, C.; Smith, G. *Aust. J. Chem.* **1990**, *43*, 1431.
- (552) Koltunova, T. K.; Samsonenko, D. G.; Dybtsev, D. N.; Fedin, V. P. *J. Struct. Chem.* **2017**, *58*, 1048–1055.
- (553) Knop, O.; Bakshi, P. K. *Can. J. Chem.* **1995**, *73*, 151–160.
- (554) Liu, Y.; Dou, J.-M.; Wang, D.-Q.; Zhang, X.-X.; Zhou, L. *Acta Crystallogr. E Struct. Rep. Online* **2006**, *62*, m2159–m2161.
- (555) Qi, X.-L. *Acta Crystallogr. E Struct. Rep. Online* **2009**, *65*, m135–m135.
- (556) Li, F.-r.; Lv, J.-h.; Yu, K.; Zhang, M.-l.; Wang, K.-p.; Meng, F.-x.; Zhou, B.-b. *CrystEngComm* **2018**, *20*, 3522–3534.
- (557) Li, P.; Goswami, S.; Otake, K.-i.; Wang, X.; Chen, Z.; Hanna, S. L.; Farha, O. K. *Inorg. Chem.* **2019**, *58*, 3586–3590.
- (558) Haitao, X.; Nengwu, Z.; Xianglin, J.; Ruyi, Y.; Yonggang, W.; Enyi, Y.; Zhengquan, L. *J. Mol. Struct.* **2003**, *655*, 339–342.
- (559) Chen, W.-T.; Luo, Q.-Y.; Liu, D.-S.; Chen, H.-L.; Xu, Y.-P. *Inorg. Chem. Commun.* **2008**, *11*, 899–902.
- (560) You, L.-X.; Zhao, B.-B.; Liu, H.-J.; Wang, S.-J.; Xiong, G.; He, Y.-K.; Ding, F.; Joos, J. J.; Smet, P. F.; Sun, Y.-G. *CrystEngComm* **2018**, *20*, 615–623.
- (561) Breeze, M. I.; Chamberlain, T. W.; Clarkson, G. J.; de Camargo, R. P.; Wu, Y.; de Lima, J. F.; Millange, F.; Serra, O. A.; O'Hare, D.; Walton, R. I. *CrystEngComm* **2017**, *19*, 2424–2433.
- (562) Zucchi, G.; Thuéry, P.; Rivière, E.; Ephritikhine, M. *Chem. Commun.* **2010**, *46*, 9143–9145.
- (563) Starynowicz, P. *J. Alloy. Compd.* **2000**, *305*, 117–120.

- (564) Starynowicz, P. *J. Alloy. Compd.* **1998**, 269, 67–70.
- (565) Cole, M. L.; Deacon, G. B.; Junk, P. C.; Proctor, K. M.; Scott, J. L.; Strauss, C. R. *Eur. J. Inorg. Chem.* **2005**, 2005, 4138–4144.
- (566) Müller-Buschbaum, K.; Mokaddem, Y.; Schappacher, F. M.; Pöttgen, R. *Angew. Chem. Int. Ed.* **2007**, 46, 4385–4387.
- (567) Khasnis, D. V.; Brewer, M.; Lee, J.; Emge, T. J.; Brennan, J. G. *J. Am. Chem. Soc.* **1994**, 116, 7129–7133.
- (568) Berardini, M.; Emge, T.; Brennan, J. G. *J. Am. Chem. Soc.* **1993**, 115, 8501–8502.
- (569) Janicki, R.; Mondry, A.; Starynowicz, P. *Z. anorg. allg. Chem.* **2005**, 631, 2475–2477.
- (570) Starynowicz, P. *Polyhedron* **1995**, 14, 3573–3577.
- (571) Plečnik, C. E.; Liu, S.; Chen, X.; Meyers, E. A.; Shore, S. G. *J. Am. Chem. Soc.* **2004**, 126, 204–213.
- (572) Müller-Buschbaum, K.; Mokaddem, Y. *Solid State Sci.* **2008**, 10, 416–420.
- (573) Gudenschwager, M.; Wickleder, M. S. CCDC 1045819: Experimental Crystal Structure Determination, 2015.
- (574) Rybak, J.-C.; Schellenberg, I.; Pöttgen, R.; Müller-Buschbaum, K. *Z. anorg. allg. Chem.* **2010**, 636, 1720–1725.
- (575) Wolf, B. M.; Stuhl, C.; Anwender, R. *Chem. Commun.* **2018**, 54, 8826–8829.
- (576) Serre, C.; Millange, F.; Marrot, J.; Férey, G. *Chem. Mater.* **2002**, 14, 2409–2415.
- (577) Starynowicz, P. *Polyhedron* **2003**, 22, 2761–2765.
- (578) Ali, S. H.; Deacon, G. B.; Junk, P. C.; Hamidi, S.; Wiecko, M.; Wang, J. *Chem. Eur. J.* **2018**, 24, 230–242.
- (579) Lee, J.; Emge, T. J.; Brennan, J. G. *Inorg. Chem.* **1997**, 36, 5064–5068.
- (580) Wu, Y.; Zheng, N.; Yang, R.; Xu, H.; Ye, E. *J. Mol. Struct.* **2002**, 610, 181–186.
- (581) Ashida, Y.; Arashiba, K.; Nakajima, K.; Nishibayashi, Y. *Nature* **2019**, 568, 536–540.
- (582) Hou, Z.; Zhang, Y.; Tezuka, H.; Xie, P.; Tardif, O.; Koizumi, T.-a.; Yamazaki, H.; Wakatsuki, Y. *J. Am. Chem. Soc.* **2000**, 122, 10533–10543.
- (583) Mashima, K.; Oshiki, T.; Tani, K. *J. Org. Chem.* **1998**, 63, 7114–7116.
- (584) Freedman, D.; Kornienko, A.; Emge, T. J.; Brennan, J. G. *Inorg. Chem.* **2000**, 39, 2168–2171.
- (585) Jaroschik, F.; Bonnet, F.; Goff, X.-F. L.; Ricard, L.; Nief, F.; Visseaux, M. *Dalton Trans.* **2010**, 39, 6761–6766.
- (586) Müller-Buschbaum, K.; Deacon, G. B.; Forsyth, C. M. *Eur. J. Inorg. Chem.* **2002**, 2002, 3172–3177.
- (587) Plečnik, C. E.; Liu, S.; Liu, J.; Chen, X.; Meyers, E. A.; Shore, S. G. *Inorg. Chem.* **2002**, 41, 4936–4943.
- (588) MacDonald, M. R.; Bates, J. E.; Ziller, J. W.; Furche, F.; Evans, W. J. *J. Am. Chem. Soc.* **2013**, 135, 9857–9868.
- (589) Cai, B.; Ren, Y.; Jiang, H.; Zheng, D.; Shi, D.; Qian, Y.; Chen, J. *CrystEngComm* **2012**, 14, 5285–5288.

- (590) Kim, B.; Khanna, R.; Koyejo, O. O. In *Advances in Neural Information Processing Systems 29*, Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., Garnett, R., Eds.; Curran Associates, Inc.: 2016, pp 2280–2288.
- (591) Garreau, D.; Jitkrittum, W.; Kanagawa, M. In *arXiv preprint Arxiv-1707.07269*, 2017.
- (592) He, H.; Garcia, E. A. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284.
- (593) Cohen, J. *Educ. Psychol. Meas.* **1960**, *20*, 37–46.
- (594) Kohavi, R. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, Morgan Kaufmann Publishers Inc.: Montreal, Quebec, Canada, 1995, pp 1137–1143.
- (595) Raschka, S. In *arXiv preprint Arxiv-1811.12808*, 2018.
- (596) Bergstra, J.; Komer, B.; Eliasmith, C.; Yamins, D.; Cox, D. D. *Comput. Sci. Disc.* **2015**, *8*, 014008.
- (597) Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. In *Advances in Neural Information Processing Systems 24*, Granada, 2011, p 10.
- (598) Putatunda, S.; Rama, K. In *Proceedings of the 2018 International Conference on Signal Processing and Machine Learning - SPML '18*, ACM Press: Shanghai, China, 2018, pp 6–10.
- (599) Komer, B.; Bergstra, J.; Eliasmith, C. In *Automated Machine Learning: Methods, Systems, Challenges*, Hutter, F., Kotthoff, L., Vanschoren, J., Eds.; The Springer Series on Challenges in Machine Learning; Springer International Publishing: Cham, 2019, pp 97–111.
- (600) Haghighi, S.; Jasemi, M.; Hessabi, S.; Zolanvari, A. *JOSS* **2018**, *3*, 729.
- (601) Byrt, T.; Bishop, J.; Carlin, J. B. *J. Clin. Epidemiol.* **1993**, *46*, 423–429.
- (602) Sindhwani, V.; Bhattacharya, P.; Rakshit, S. In *Proceedings of the 2001 SIAM International Conference on Data Mining*, Society for Industrial and Applied Mathematics: 2001, pp 1–18.
- (603) Matthews, B. *Biochim. Biophys. Acta - Proteins Proteom.* **1975**, *405*, 442–451.
- (604) Bekkar, M.; Djemaa, D. H. K. *Journal of Information Engineering and Applications* **2013**, *13*.
- (605) Efron, B.; Tibshirani, R. *Statist. Sci.* **1986**, *1*, 54–75.
- (606) Ojala, M.; Garriga, G. C. In *2009 Ninth IEEE International Conference on Data Mining*, IEEE: Miami Beach, FL, USA, 2009, pp 908–913.
- (607) Good, P., *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*; Springer Series in Statistics; Springer-Verlag: New York, 1994.
- (608) Chuang, K. V.; Keiser, M. J. *Science* **2018**, *362*, eaat8603.
- (609) Niculescu-Mizil, A.; Caruana, R. In *Proceedings of the 22nd International Conference on Machine Learning - ICML '05*, ACM Press: Bonn, Germany, 2005, pp 625–632.
- (610) Bröcker, J.; Smith, L. A. *Wea. Forecasting* **2007**, *22*, 651–661.
- (611) Mehta, P.; Bukov, M.; Wang, C.-H.; Day, A. G. R.; Richardson, C.; Fisher, C. K.; Schwab, D. J. *Phys. Rep.* **2019**, *810*, 1–124.
- (612) Ho, J.; Tumkaya, T.; Aryal, S.; Choi, H.; Claridge-Chang, A. *Nat. Methods* **2019**, *16*, 565–566.

- (613) Taube, H.; Myers, H.; Rich, R. L. *J. Am. Chem. Soc.* **1953**, *75*, 4118–4119.
- (614) Kaim, W.; Klein, A.; Glöckle, M. *Acc. Chem. Res.* **2000**, *33*, 755–763.
- (615) Ward, L.; Agrawal, A.; Choudhary, A.; Wolverton, C. *npj Comput. Mater.* **2016**, *2*.
- (616) Steinhardt, P. J.; Nelson, D. R.; Ronchetti, M. *Phys. Rev. B* **1983**, *28*, 784–805.
- (617) Behler, J. *J. Chem. Phys.* **2011**, *134*, 074106.
- (618) Broto, P.; Moreau, G.; Vandycke, C. *Eur. J. Med. Chem.* **1984**, *19*, 71–78.
- (619) Ioannidis, E. I.; Gani, T. Z. H.; Kulik, H. J. *J. Comput. Chem.* **2016**, *37*, 2106–2117.
- (620) In *The Organometallic Chemistry of the Transition Metals*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2005, pp 1–28.
- (621) Burrows, A., *Chemistry³: Introducing Inorganic, Organic and Physical Chemistry*, Third edition; Oxford University Press: Oxford, 2017.
- (622) Yue, Y.; Arman, H.; Tonzetich, Z. J.; Chen, B. *Z. anorg. allg. Chem.* **2019**, *645*, 797–800.
- (623) Storr, T.; Thompson, J. R.; Patrick, B. O.; Reiff, W. M.; Storr, A.; Thompson, R. C. *Polyhedron* **2016**, *108*, 80–86.
- (624) Kumar, S.; Arora, A.; Kaushal, J.; Oswal, P.; Kumar, A.; Tomar, K. *New J. Chem.* **2019**, *43*, 4338–4341.
- (625) Rosi, N. L.; Kim, J.; Eddaoudi, M.; Chen, B.; O’Keeffe, M.; Yaghi, O. M. *J. Am. Chem. Soc.* **2005**, *127*, 1504–1518.
- (626) Lammert, M.; Wharmby, M. T.; Smolders, S.; Bueken, B.; Lieb, A.; Lomachenko, K. A.; Vos, D. D.; Stock, N. *Chem. Commun.* **2015**, *51*, 12578–12581.
- (627) Febriansyah, B.; Koh, T. M.; John, R. A.; Ganguly, R.; Li, Y.; Bruno, A.; Mhaisalkar, S. G.; England, J. *Chem. Mater.* **2018**, *30*, 5827–5830.
- (628) Atzori, C.; Lomachenko, K. A.; Øien-Ødegaard, S.; Lamberti, C.; Stock, N.; Barolo, C.; Bonino, F. *Cryst. Growth Des.* **2019**, *19*, 787–796.
- (629) Wong-Ng, W.; Kaduk, J. A.; Wu, H.; Suchomel, M. *Powder Diffr.* **2012**, *27*, 256–262.
- (630) Ji, G.; Liu, J.; Gao, X.; Sun, W.; Wang, J.; Zhao, S.; Liu, Z. *J. Mater. Chem. A* **2017**, *5*, 10200–10205.
- (631) Baron, M.; Tubaro, C.; Basato, M.; Biffis, A.; Natile, M. M.; Graiff, C. *Organometallics* **2011**, *30*, 4607–4615.
- (632) Chen, X.-Y.; Wei, R.-J.; Zheng, L.-S.; Tao, J. *Inorg. Chem.* **2014**, *53*, 13212–13219.
- (633) Zeng, M.-H.; Yin, Z.; Tan, Y.-X.; Zhang, W.-X.; He, Y.-P.; Kurmoo, M. *J. Am. Chem. Soc.* **2014**, *136*, 4680–4688.
- (634) Falaise, C.; Volkringer, C.; Vigier, J.-F.; Henry, N.; Beaurain, A.; Loiseau, T. *Chem. Eur. J.* **2013**, *19*, 5324–5331.
- (635) Dong, X.-Y.; Wang, R.; Li, J.-B.; Zang, S.-Q.; Hou, H.-W.; Mak, T. C. W. *Chem. Commun.* **2013**, *49*, 10590–10592.
- (636) Bai, L.; Wang, H.-B.; Li, D.-S.; Wu, Y.-P.; Zhao, J.; Ma, L.-F. *Inorg. Chem. Commun.* **2014**, *44*, 188–190.
- (637) Goswami, S.; Adhikary, A.; Jena, H. S.; Biswas, S.; Konar, S. *Inorg. Chem.* **2013**, *52*, 12064–12069.

- (638) Bernini, M. C.; de Paz, J. R.; Snejko, N.; Sáez-Puche, R.; Gutierrez-Puebla, E.; Monge, M. Á. *Inorg. Chem.* **2014**, *53*, 12885–12895.
- (639) Liu, G.-Z.; Li, X.-L.; Xin, L.-Y.; Wang, L.-Y. *CrystEngComm* **2012**, *14*, 5315–5321.
- (640) Livage, C.; Egger, C.; Férey, G. *Chem. Mater.* **1999**, *11*, 1546–1550.
- (641) Wang, X.-W.; Dong, Y.-R.; Zheng, Y.-Q.; Chen, J.-Z. *Cryst. Growth Des.* **2007**, *7*, 613–615.
- (642) Spirkel, S.; Grzywa, M.; Reschke, S.; Fischer, J. K. H.; Sippel, P.; Demeshko, S.; Krug von Nidda, H.-A.; Volkmer, D. *Inorg. Chem.* **2017**, *56*, 12337–12347.
- (643) Jana, S.; Chattopadhyay, S. *J. Coord. Chem.* **2013**, *66*, 3906–3914.
- (644) Zhang, T.-Z.; Zhang, Z.-M.; Lu, Y.; Wang, E.-B. *J. Coord. Chem.* **2012**, *65*, 48–54.
- (645) Zhu, X.; Zhao, S.; Peng, Y.-F.; Li, B.-L.; Wu, B. *CrystEngComm* **2013**, *15*, 9154–9160.
- (646) Wen, R.-M.; Han, S.-D.; Wang, H.; Zhang, Y.-H. *Chinese Chem. Lett.* **2014**, *25*, 854–858.
- (647) Bouketaya, S.; Smida, M.; Abdelbaky, M. S. M.; Dammak, M.; Garcia-Granda, S. *J. Solid State Chem.* **2018**, *262*, 343–350.
- (648) Liu, L.; A. DeGayner, J.; Sun, L.; Z. Zee, D.; David Harris, T. *Chem. Sci.* **2019**, *10*, 4652–4661.
- (649) Vlad, A.; Zaltariov, M.-F.; Shova, S.; Novitchi, G.; Varganici, C.-D.; Train, C.; Cazacu, M. *CrystEngComm* **2013**, *15*, 5368–5375.
- (650) Gong, T.; Yang, X.; Fang, J.-J.; Sui, Q.; Xi, F.-G.; Gao, E.-Q. *ACS Appl. Mater. Interfaces* **2017**, *9*, 5503–5512.
- (651) Zhang, S.; Liu, X.; Liu, B.; Xia, Z.; Wang, W.; Yang, Q.; Ke, H.; Wei, Q.; Xie, G.; Chen, S.; Gao, S. *Sci. China Chem.* **2015**, *58*, 1032–1038.
- (652) Liao, L.; Ingram, C. W.; Vandever, D.; Hardcastle, K.; Solntsev, K. M.; Sabo, D.; John Zhang, Z.; Weber, R. T. *Inorg. Chim. Acta* **2012**, *391*, 1–9.
- (653) Choi, S. B.; Seo, M. J.; Cho, M.; Kim, Y.; Jin, M. K.; Jung, D.-Y.; Choi, J.-S.; Ahn, W.-S.; Rowsell, J. L. C.; Kim, J. *Cryst. Growth Des.* **2007**, *7*, 2290–2293.
- (654) Šimėnas, M.; Kobalz, M.; Mendt, M.; Eckold, P.; Krautscheid, H.; Banys, J.; Pöppel, A. *J. Phys. Chem. C* **2015**, *119*, 4898–4907.
- (655) Ghosh, S. K.; Ribas, J.; Bharadwaj, P. K. *CrystEngComm* **2004**, *6*, 250–256.
- (656) Ding, X.; Chen, L.; Honsho, Y.; Feng, X.; Saengsawang, O.; Guo, J.; Saeki, A.; Seki, S.; Irle, S.; Nagase, S.; Parasuk, V.; Jiang, D. *J. Am. Chem. Soc.* **2011**, *133*, 14510–14513.
- (657) Ding, X.; Feng, X.; Saeki, A.; Seki, S.; Nagai, A.; Jiang, D. *Chem. Commun.* **2012**, *48*, 8952.
- (658) Nagai, A.; Chen, X.; Feng, X.; Ding, X.; Guo, Z.; Jiang, D. *Angew. Chem. Int. Ed.* **2013**, *52*, 3770–3774.
- (659) Jin, S.; Supur, M.; Addicoat, M.; Furukawa, K.; Chen, L.; Nakamura, T.; Fukuzumi, S.; Irle, S.; Jiang, D. *J. Am. Chem. Soc.* **2015**, *137*, 7817–7827.
- (660) Diercks, C. S.; Lin, S.; Kornienko, N.; Kapustin, E. A.; Nichols, E. M.; Zhu, C.; Zhao, Y.; Chang, C. J.; Yaghi, O. M. *J. Am. Chem. Soc.* **2018**, *140*, 1116–1122.

- (661) Vardhan, H.; Hou, L.; Yee, E.; Nafady, A.; Al-Abdrabalnabi, M. A.; Al-Enizi, A. M.; Pan, Y.; Yang, Z.; Ma, S. *ACS Sustainable Chem. Eng.* **2019**, *7*, 4878–4888.
- (662) Lu, M.; Liu, J.; Li, Q.; Zhang, M.; Liu, M.; Wang, J.-L.; Yuan, D.-Q.; Lan, Y.-Q. *Angew. Chem. Int. Ed.* **2019**, *58*, 12392–12397.
- (663) Smolders, S.; Lomachenko, K. A.; Bueken, B.; Struyf, A.; Bugaev, A. L.; Atzori, C.; Stock, N.; Lamberti, C.; Roeffaers, M. B. J.; De Vos, D. E. *ChemPhysChem* **2018**, *19*, 373–378.
- (664) Cliffe, M. J.; Wan, W.; Zou, X.; Chater, P. A.; Kleppe, A. K.; Tucker, M. G.; Wilhelm, H.; Funnell, N. P.; Coudert, F.-X.; Goodwin, A. L. *Nat. Commun.* **2014**, *5*, 4176.
- (665) Cliffe, M. J.; Hill, J. A.; Murray, C. A.; Coudert, F.-X.; Goodwin, A. L. *Phys. Chem. Chem. Phys.* **2015**, *17*, 11586–11592.
- (666) Liu, L.; Chen, Z.; Wang, J.; Zhang, D.; Zhu, Y.; Ling, S.; Huang, K.-W.; Belmabkhout, Y.; Adil, K.; Zhang, Y.; Slater, B.; Eddaoudi, M.; Han, Y. *Nat. Chem.* **2019**, *11*, 622–628.
- (667) Trickett, C. A.; Gagnon, K. J.; Lee, S.; Gándara, F.; Bürgi, H.-B.; Yaghi, O. M. *Angew. Chem. Int. Ed.* **2015**, *54*, 11162–11167.
- (668) Ling, S.; Slater, B. *Chem. Sci.* **2016**, *7*, 4706–4712.
- (669) Svane, K. L.; Bristow, J. K.; Gale, J. D.; Walsh, A. J. *Mater. Chem. A* **2018**, *6*, 8507–8513.
- (670) Drake, H. F.; Day, G. S.; Vali, S. W.; Xiao, Z.; Banerjee, S.; Li, J.; Joseph, E. A.; Kuszynski, J. E.; Perry, Z. T.; Kirchon, A.; Ozdemir, O. K.; Lindahl, P. A.; Zhou, H.-C. *Chem. Commun.* **2019**, *55*, 12769–12772.
- (671) Thapa, S.; Hettiarachchi, E.; Dickie, D. A.; Rubasinghege, G.; Qin, Y. *Chem. Commun.* **2018**, *54*, 12654–12657.
- (672) Xu, L.; Liu, B.; Liu, S.-X.; Jiao, H.; de Castro, B.; Cunha-Silva, L. *CrystEngComm* **2014**, *16*, 10649–10657.
- (673) Nättinen, K. I.; Rissanen, K. *Inorg. Chem.* **2003**, *42*, 5126–5134.
- (674) Eubank, J. F.; Nouar, F.; Luebke, R.; Cairns, A. J.; Wojtas, L.; Alkordi, M.; Bousquet, T.; Hight, M. R.; Eckert, J.; Embs, J. P.; Georgiev, P. A.; Eddaoudi, M. *Angew. Chem. Int. Ed.* **2012**, *51*, 10099–10103.
- (675) Huo, M.; Yang, W.; Zhang, H.; Zhang, L.; Liao, J.; Lin, L.; Lu, C. *RSC Adv.* **2016**, *6*, 111549–111555.
- (676) Song, J.; Luo, Z.; Britt, D. K.; Furukawa, H.; Yaghi, O. M.; Hardcastle, K. I.; Hill, C. L. *J. Am. Chem. Soc.* **2011**, *133*, 16839–16846.
- (677) Bai, H.-Y.; Yang, J.; Liu, B.; Ma, J.-F.; Kan, W.-Q.; Liu, Y.-Y.; Liu, Y.-Y. *CrystEngComm* **2011**, *13*, 5877–5884.
- (678) Li, X.-Q.; Zhang, H.-B.; Wu, S.-T.; Lin, J.-D.; Lin, P.; Li, Z.-H.; Du, S.-W. *CrystEngComm* **2012**, *14*, 936–944.
- (679) Van Albada, G. A.; Ghazzali, M.; Al-Farhan, K.; Mutikainen, I.; Reedijk, J. *Inorg. Chem. Commun.* **2011**, *14*, 162–165.
- (680) Chaudhari, A. K.; Nagarkar, S. S.; Joarder, B.; Ghosh, S. K. *Cryst. Growth Des.* **2013**, *13*, 3716–3721.
- (681) Brown, K.; Zolezzi, S.; Aguirre, P.; Venegas-Yazigi, D.; Paredes-Garcia, V.; Baggio, R.; A. Novak, M.; Spodine, E. *Dalton T.* **2009**, *0*, 1422–1427.

- (682) Park, J.; Hinckley, A. C.; Huang, Z.; Feng, D.; Yakovenko, A. A.; Lee, M.; Chen, S.; Zou, X.; Bao, Z. *J. Am. Chem. Soc.* **2018**, *140*, 14533–14537.
- (683) Wang, X.; Wang, Y.; Liu, G.; Tian, A.; Zhang, J.; Lin, H. *Dalton T.* **2011**, *40*, 9299–9305.
- (684) Yamabayashi, T.; Atzori, M.; Tesi, L.; Cosquer, G.; Santanni, F.; Boulon, M.-E.; Morra, E.; Benci, S.; Torre, R.; Chiesa, M.; Sorace, L.; Sessoli, R.; Yamashita, M. *J. Am. Chem. Soc.* **2018**, *140*, 12090–12101.
- (685) Tong, J.; Demeshko, S.; John, M.; Dechert, S.; Meyer, F. *Inorg. Chem.* **2016**, *55*, 4362–4372.
- (686) Birk, T.; Pedersen, K. S.; Thuesen, C. A.; Weyhermüller, T.; Schau-Magnussen, M.; Piligkos, S.; Weihe, H.; Mossin, S.; Evangelisti, M.; Bendix, J. *Inorg. Chem.* **2012**, *51*, 5435–5443.
- (687) Wong, J. W. L.; Demeshko, S.; Dechert, S.; Meyer, F. *Inorg. Chem.* **2019**, *58*, 13337–13345.
- (688) Shiga, T.; Sato, Y.; Tachibana, M.; Sato, H.; Matsumoto, T.; Sagayama, H.; Kumai, R.; Murakami, Y.; Newton, G. N.; Oshio, H. *Inorg. Chem.* **2018**, *57*, 14013–14017.
- (689) Fei, H.; Han, C. S.; Robins, J. C.; Oliver, S. R. *J. Chem. Mater.* **2013**, *25*, 647–652.
- (690) Skelton, J. Transformer, <https://github.com/JMSkelton/Transformer>, accessed 2023-3-9, 2020.
- (691) O’Keefe, M.; Brese, N. E. *J. Am. Chem. Soc.* **1991**, *113*, 3226–3229.
- (692) Tarassoli, A.; Nobakht, V.; Baladi, E.; Carlucci, L.; Proserpio, D. M. *CrystEngComm* **2017**, *19*, 6116–6126.
- (693) Patra, G. K.; Goldberg, I. *J. Chem. Soc., Dalton Trans.* **2002**, 1051–1057.
- (694) Sun, X.; You, W.; Cheng, H.; Zhang, F.; Meng, B.; Zhang, L. *Inorg. Chim. Acta* **2011**, *373*, 137–141.
- (695) Ouellette, W.; Burkholder, E.; Manzar, S.; Bewley, L.; Rarig, R. S.; Zubieta, J. *Solid State Sci.* **2004**, *6*, 77–84.
- (696) Wang, Y.; He, C.-T.; Liu, Y.-J.; Zhao, T.-Q.; Lu, X.-M.; Zhang, W.-X.; Zhang, J.-P.; Chen, X.-M. *Inorg. Chem.* **2012**, *51*, 4772–4778.
- (697) Vakulka, A.; Goreshnik, E. *J. Coord. Chem.* **2018**, *71*, 2426–2440.
- (698) Nagarkar, S. S.; Kurasho, H.; Duong, N. T.; Nishiyama, Y.; Kitagawa, S.; Horike, S. *Chem. Commun.* **2019**, *55*, 5455–5458.
- (699) Bermejo, M. R.; Fondo, M.; Garcia-Deibe, A.; González, A. M.; Sousa, A.; Sanmartin, J.; McAuliffe, C. A.; Pritchard, R. G.; Watkinson, M.; Lukov, V. *Inorg. Chim. Acta* **1999**, *293*, 210–217.
- (700) Tazelaar, C. G. J.; Nicolas, E.; van Dijk, T.; Broere, D. L. J.; Cardol, M.; Lutz, M.; Gudat, D.; Slootweg, J. C.; Lammertsma, K. *Dalton Trans.* **2016**, *45*, 2237–2249.
- (701) Fang, W.-H.; Yang, G.-Y. *CrystEngComm* **2013**, *15*, 9504.
- (702) Li, X.-Z.; Zhou, X.-P.; Li, D.; Yin, Y.-G. *CrystEngComm* **2011**, *13*, 6759.
- (703) Thomas, J.; Ramanan, A. *J. Chem. Sci.* **2016**, *128*, 1687–1694.
- (704) Banerjee, D.; Kim, S. J.; Wu, H.; Xu, W.; Borkowski, L. A.; Li, J.; Parise, J. B. *Inorg. Chem.* **2011**, *50*, 208–212.
- (705) Näther, C.; Jeß, I.; Bolte, M. *Z. Naturforsch. B* **2003**, *58*, 1105–1111.

- (706) Roger, M.; Arliguie, T.; Thuéry, P.; Fourmigué, M.; Ephritikhine, M. *Inorg. Chem.* **2005**, *44*, 594–600.
- (707) Yan, B.; Golub, V. O.; Lachgar, A. *Inorg. Chim. Acta* **2006**, *359*, 118–126.
- (708) Dobrott, R. D.; Lipscomb, W. N. *J. Chem. Phys.* **1962**, *37*, 1779–1784.
- (709) Mak, T.; Yip, W.; Kennard, C.; Smith, G.; Oreilly, E. *Aust. J. Chem.* **1988**, *41*, 683.
- (710) Chhetri, P. M.; Chang, Y.-C.; Hu, H.-L.; Chiang, Y.-H.; Yang, X.-K.; Chen, J.-D. *J. Mol. Struct.* **2017**, *1144*, 173–180.
- (711) Chorazy, S.; Kumar, K.; Nakabayashi, K.; Sieklucka, B.; Ohkoshi, S.-i. *Inorg. Chem.* **2017**, *56*, 5239–5252.
- (712) Huerta, R.; Flores-Figueroa, A.; Ugalde-Saldivar, V. M.; Castillo, I. *Inorg. Chem.* **2007**, *46*, 9510–9512.
- (713) Dunning, S. G.; Reynolds, J. E.; Walsh, K. M.; Kristek, D. J.; Lynch, V. M.; Kunal, P.; Humphrey, S. M. *Organometallics* **2019**, *38*, 3406–3411.
- (714) Lal, G.; Gelfand, B. S.; Lin, J.-B.; Banerjee, A.; Trudel, S.; Shimizu, G. K. H. *Inorg. Chem.* **2019**, *58*, 9874–9881.
- (715) Hu, Y.-Q.; Zhang, T.; Li, M.-Q.; Wang, Y.; Zheng, Z.; Zheng, Y.-Z. *Green Chem.* **2017**, *19*, 1250–1254.
- (716) Wang, J.; Tong, M.-L. CCDC 262599: Experimental Crystal Structure Determination, 2013.
- (717) Yang, X.-J.; Li, H.-X.; Xu, Z.-L.; Li, H.-Y.; Ren, Z.-G.; Lang, J.-P. *CrystEngComm* **2012**, *14*, 1641–1652.
- (718) Chakrabarty, P. P.; Saha, S.; Schollmeyer, D.; Boudalis, A. K.; Jana, A. D.; Luneau, D. *J. Coord. Chem.* **2013**, *66*, 9–17.
- (719) Li, W.; Li, M.-X.; Shao, M.; Wang, Z.-X.; Liu, H.-J. *Inorg. Chem. Commun.* **2008**, *11*, 954–957.
- (720) Tandon, S. S.; Thompson, L. K.; Bridson, J. N.; McKee, V.; Downard, A. J. *Inorg. Chem.* **1992**, *31*, 4635–4642.
- (721) Lu, L.-R.; Qi, C.; Wang, Z.-X.; He, X.; Li, M.-X. *Inorg. Nano-Met. Chem.* **2017**, *47*, 1248–1253.
- (722) Zhang, S.-S.; Zhan, S.-Z.; Li, M.; Peng, R.; Li, D. *Inorg. Chem.* **2007**, *46*, 4365–4367.
- (723) Agarwal, R. A.; Aijaz, A.; Sañudo, C.; Xu, Q.; Bharadwaj, P. K. *Cryst. Growth Des.* **2013**, *13*, 1238–1245.
- (724) Chen, Z.; Zuo, Y.; Li, X.-H.; Wang, H.; Zhao, B.; Shi, W.; Cheng, P. *J. Mol. Struct.* **2008**, *888*, 360–365.
- (725) Meghdadi, S.; Amirnasr, M.; Yavari, E.; Mereiter, K.; Bagheri, M. *Cr. Chim.* **2017**, *20*, 730–737.
- (726) Di Nicola, C.; Effendy; Pettinari, C.; Skelton, B. W.; Somers, N.; White, A. H. *Inorg. Chim. Acta* **2006**, *359*, 53–63.
- (727) Puttreddy, R.; Peuronen, A.; Lahtinen, M.; Rissanen, K. *Cryst. Growth Des.* **2019**, *19*, 3815–3824.
- (728) Koziel, M.; Pelka, R.; Rams, M.; Nitek, W.; Sieklucka, B. *Inorg. Chem.* **2010**, *49*, 4268–4277.
- (729) Nockemann, P.; Meyer, G. *Z. anorg. allg. Chem.* **2003**, *629*, 1294–1299.

- (730) Chen, L.-J.; He, X.; Xia, C.-K.; Zhang, Q.-Z.; Chen, J.-T.; Yang, W.-B.; Lu, C.-Z. *Cryst. Growth Des.* **2006**, *6*, 2076–2085.
- (731) Atherton, Z.; Goodgame, D. M. L.; Menzer, S.; Williams, D. J. *Inorg. Chem.* **1998**, *37*, 849–858.
- (732) Senchyk, G. A.; Lysenko, A. B.; Domasevitch, K. V.; Erhart, O.; Henfling, S.; Krautscheid, H.; Rusanov, E. B.; Krämer, K. W.; Decurtins, S.; Liu, S.-X. *Inorg. Chem.* **2017**, *56*, 12952–12966.
- (733) Fu, W.-W.; Chen, Z.-N. *Acta Crystallogr. E Struct. Rep. Online* **2007**, *63*, m842–m844.
- (734) Liu, W.; Banerjee, D.; Lin, F.; Li, J. *J. Mater. Chem. C* **2019**, *7*, 1484–1490.
- (735) Weber, R.; Bergerhoff, G. *Z. Kristallogr. Cryst. Mater.* **1991**, *195*.
- (736) Romero, I.; Sánchez-Castelló, G.; Teixidor, F.; Whitaker, C. R.; Rius, J.; Miravittles, C.; Flor, T.; Escriche, L.; Casabó, J. *Polyhedron* **1996**, *15*, 2057–2065.
- (737) Barclay, G. A.; Vagg, R. S.; Watton, E. C. *Acta Crystallogr. B Struct. Sci* **1978**, *34*, 1833–1837.
- (738) Fang, R.-Q.; Zhao, Y.-F.; Zhang, X.-M. *Inorg. Chim. Acta* **2006**, *359*, 2023–2028.
- (739) You, Y. S.; Yoon, J. H.; Lim, J. H.; Kim, H. C.; Hong, C. S. *Inorg. Chim. Acta* **2007**, *360*, 2523–2531.
- (740) Finn, R. C.; Sims, J.; O'Connor, C. J.; Zubieta, J. *J. Chem. Soc., Dalton Trans.* **2002**, 159.
- (741) Luo, J.; Alexander, B.; Wagner, T. R.; Maggard, P. A. *Inorg. Chem.* **2004**, *43*, 5537–5542.
- (742) Jiang, L.; Feng, X.-L.; Lu, T.-B.; Gao, S. *Inorg. Chem.* **2006**, *45*, 5018–5026.
- (743) Chen, L.; Thompson, L. K.; Bridson, J. N. *Can. J. Chem.* **1992**, *70*, 2709–2716.
- (744) Gong, Y.; Wu, T.; Lin, J. *CrystEngComm* **2012**, *14*, 3727.
- (745) Yu, K.; Chen, W.-L.; Zhou, B.-B.; Li, Y.-G.; Yu, Y.; Su, Z.-H.; Gao, S.; Chen, Y. *CrystEngComm* **2011**, *13*, 3417–3424.
- (746) Bhunia, A.; Lan, Y.; Mereacre, V.; Gamer, M. T.; Powell, A. K.; Roesky, P. W. *Inorg. Chem.* **2011**, *50*, 12697–12704.
- (747) Shen, H.-Y.; Liao, D.-Z.; Jiang, Z.-H.; Yan, S.-P.; Sun, B.-W.; Wang, G.-L.; Xin-Kan, Y.; Wang, H.-G. *Chem. Lett.* **1998**, *27*, 469–470.
- (748) Pedersen, K. S. et al. *Nat. Chem.* **2018**, *10*, 1056–1061.
- (749) Brodersen, K.; Knoerr, A. Z. *Naturforsch. B* **1990**, *45*, 1193.
- (750) Song, X.; Yang, P.; Mei, X.; Li, L.; Liao, D. *Eur. J. Inorg. Chem.* **2010**, *2010*, 1689–1695.
- (751) Zhang, X.-F.; Gao, S.; Huo, L.-H.; Ng, S. W. *Acta Cryst. E* **2006**, *62*, m3418–m3419.
- (752) Wang, R.-H.; Hong, M.-C.; Luo, J.-H.; Cao, R.; Weng, J.-B. *Eur. J. Inorg. Chem.* **2002**, *2002*, 2082–2085.
- (753) Klausmeyer, K. K.; Beckles, F. R. *Inorg. Chim. Acta* **2007**, *360*, 3241–3249.
- (754) Huang, Y.-G.; Mu, B.; Schoenecker, P. M.; Carson, C. G.; Karra, J. R.; Cai, Y.; Walton, K. S. *Angew. Chem. Int. Ed.* **2011**, *50*, 436–440.
- (755) Choi, H. J.; Suh, M. P. *J. Am. Chem. Soc.* **2004**, *126*, 15844–15851.

- (756) Behrsing, T.; Deacon, G. B.; Forsyth, C. M.; Forsyth, M.; Skelton, B. W.; White, A. H. *Z. anorg. allg. Chem.* **2003**, 629, 35–44.
- (757) Shen, X.-P.; Li, Y.-Z.; Yuan, A.-H.; Xu, Z. *J. Mol. Struct.* **2005**, 754, 106–110.
- (758) Yoshida, Y.; Inoue, K.; Kurmoo, M. *Inorg. Chem.* **2009**, 48, 10726–10736.
- (759) Levy, C. J.; Vittal, J. J.; Puddephatt, R. J. *Organometallics* **1996**, 15, 35–42.
- (760) Peters, K.; Peters, E.-M.; von Schnering, H. G.; Abicht, H.-P. *Z. Kristallogr. Cryst. Mater.* **1985**, 171.
- (761) Decurtins, S.; Schmalle, H. W.; Pellaux, R.; Schneuwly, P.; Hauser, A. *Inorg. Chem.* **1996**, 35, 1451–1460.
- (762) Matoga, D.; Roztocki, K.; Wilke, M.; Emmerling, F.; Oszejca, M.; Fitta, M.; Bałanda, M. *CrystEngComm* **2017**, 19, 2987–2995.
- (763) Champness, N. R.; Levason, W.; Preece, S. R.; Webster, M.; Frampton, C. S. *Inorg. Chim. Acta* **1996**, 244, 65–72.
- (764) Turpeinen, U.; Ahlgrén, M.; Hämäläinen, R. *Acta Crystallogr. B Struct. Sci.* **1982**, 38, 1580–1583.
- (765) Tzeng, B.-C.; Lin, J.-F. *Dalton Trans.* **2019**, 48, 4046–4057.
- (766) Ama, T.; Miyazaki, J.-i.; Hamada, K.; Okamoto, K.-i.; Yonemura, T.; Kawaguchi, H.; Yasui, T. *Chem. Lett.* **1995**, 24, 267–268.
- (767) Liang, X.; Parkinson, J. A.; Parsons, S.; Weishäupl, M.; Sadler, P. J. *Inorg. Chem.* **2002**, 41, 4539–4547.
- (768) Li, B.; Chen, L.-Q.; Wei, R.-J.; Tao, J.; Huang, R.-B.; Zheng, L.-S.; Zheng, Z. *Inorg. Chem.* **2011**, 50, 424–426.
- (769) Punji, B.; Mague, J. T.; Balakrishna, M. S. *Eur. J. Inorg. Chem.* **2007**, 2007, 720–731.
- (770) He, H.; Dubey, M.; Sykes, A. G.; May, P. S. *Dalton Trans.* **2010**, 39, 6466–6474.
- (771) Dutta, G.; Mandal, D.; Gupta, B. D. *J. Organomet. Chem.* **2012**, 706–707, 30–36.
- (772) Davidson, G. J. E.; Sharma, S.; Loeb, S. J. *Angew. Chem. Int. Ed.* **2010**, 49, 4938–4942.
- (773) Hou, B.-W.; Li, K. *Wuji Huaxue Xuebao* **2017**, 33, 1007.
- (774) Gemel, C.; Kickelbick, G.; Schmid, R.; Kirchner, K. *J. Chem. Soc., Dalton Trans.* **1997**, 2113–2118.
- (775) Spetzler, V.; Rijnberk, H.; Näther, C.; Bensch, W. *Z. anorg. allg. Chem.* **2004**, 630, 142–148.
- (776) Bonnitche, P. D.; Kim, B. J.; Hocking, R. K.; Clegg, J. K.; Turner, P.; Neville, S. M.; Hambley, T. W. *Dalton Trans.* **2012**, 41, 11293–11304.
- (777) Masciocchi, N.; Ardizzoia, G. A.; Brenna, S.; LaMonica, G.; Maspero, A.; Galli, S.; Sironi, A. *Inorg. Chem.* **2002**, 41, 6080–6089.
- (778) Sicilia, V.; Forniés, J.; Casas, J. M.; Martín, A.; López, J. A.; Larraz, C.; Borja, P.; Ovejero, C.; Tordera, D.; Bolink, H. *Inorg. Chem.* **2012**, 51, 3427–3435.
- (779) Kumar, G.; Gupta, R. *Inorg. Chem.* **2013**, 52, 10773–10787.
- (780) Coucouvanis, D.; Draganjac, M. E.; Koo, S. M.; Toupadakis, A.; Hadjikyriacou, A. I. *Inorg. Chem.* **1992**, 31, 1186–1196.
- (781) Petrovic, D.; Bannenberg, T.; Randoll, S.; Jones, P. G.; Tamm, M. *Dalton Trans.* **2007**, 2812–2822.

- (782) Murtaza, S.; Butler, P.; Kratky, C.; Gruber, K.; Kräutler, B. *Chem. Eur. J.* **2008**, *14*, 7521–7524.
- (783) Schneider, W.; Bauer, A.; Schier, A.; Schmidbaur, H. *Chem. Ber./Recl.* **1997**, *130*, 1423–1426.
- (784) Eid, S.; Fourmigué, M.; Roisnel, T.; Lorcy, D. *Inorg. Chem.* **2007**, *46*, 10647–10654.
- (785) Xie, L.-M.; Wang, G.-P.; He, H.-Y.; Zhu, L.-G. *Z. Kristallogr. - New Cryst. Struct.* **2003**, *218*, 245–246.
- (786) Casas, J. S.; Castellano, E. E.; Couce, M. D.; Ellena, J.; Sánchez, A.; Sordo, J.; Taboada, C. *J. Inorg. Biochem.* **2006**, *100*, 1858–1860.
- (787) Kitadai, K.; Takahashi, M.; Takeda, M.; Bhargava, S. K.; Privér, S. H.; Bennett, M. A. *Dalton Trans.* **2006**, 2560–2571.
- (788) Lee, H.-H.; Park, I.-H.; Kim, S.; Lee, E.; Ju, H.; Jung, J. H.; Ikeda, M.; Habata, Y.; Lee, S. S. *Chem. Sci.* **2017**, *8*, 2592–2596.
- (789) Muller, E.; Bernardinelli, G.; Reedijk, J. *Inorg. Chem.* **1995**, *34*, 5979–5988.
- (790) Moore, P.; Errington, W.; Sangha, S. P. *Helv. Chim. Acta* **2005**, *88*, 782–795.
- (791) Yin, G.-Q.; Wei, Q.-H.; Zhang, L.-Y.; Chen, Z.-N. *Organometallics* **2006**, *25*, 580–587.
- (792) Holler, S.; Tüchler, M.; Steller, B. G.; Belaj, F.; Veiros, L. F.; Kirchner, K.; Mösch-Zanetti, N. C. *Inorg. Chem.* **2018**, *57*, 6921–6931.
- (793) Hofreiter, S.; Paul, M.; Schmidbaur, H. *Chem. Ber.* **1995**, *128*, 901–905.
- (794) Knapp, S.; Keenan, T. P.; Zhang, X.; Fikar, R.; Potenza, J. A.; Schugar, H. J. *J. Am. Chem. Soc.* **1990**, *112*, 3452–3464.
- (795) Jeffery, J. C.; Riis-Johannessen, T.; Anderson, C. J.; Adams, C. J.; Robinson, A.; Argent, S. P.; Ward, M. D.; Rice, C. R. *Inorg. Chem.* **2007**, *46*, 2417–2426.
- (796) Shi, M.; Jiang, J.-K. *Chirality* **2003**, *15*, 605–608.
- (797) Davies, M. K.; Raithby, P. R.; Rennie, M.-A.; Steiner, A.; Wright, D. S. *J. Chem. Soc., Dalton Trans.* **1995**, 2707–2709.
- (798) Wang, Y.; Peng, Y.; Xiao, L.-N.; Hu, Y.-Y.; Wang, L.-M.; Gao, Z.-M.; Wang, T.-G.; Wu, F.-Q.; Cui, X.-B.; Xu, J.-Q. *CrystEngComm* **2012**, *14*, 1049–1056.
- (799) Chivers, T.; Clark, T. J.; Krahn, M.; Parvez, M.; Schatte, G. *Eur. J. Inorg. Chem.* **2003**, *2003*, 1857–1860.
- (800) Ohta, T.; Tachiyama, T.; Yoshizawa, K.; Yamabe, T.; Uchida, T.; Kitagawa, T. *Inorg. Chem.* **2000**, *39*, 4358–4369.
- (801) Zhu, T.-T.; Wang, J.; Chen, S.-H. *J. Mol. Struct.* **2017**, *1149*, 766–770.
- (802) Judmaier, M. E.; Holzer, C.; Volpe, M.; Mösch-Zanetti, N. C. *Inorg. Chem.* **2012**, *51*, 9956–9966.
- (803) Igashira-Kamiyama, A.; Saito, M.; Konno, T. *Chem. Lett.* **2009**, *38*, 1168–1169.
- (804) Sun, A.-H.; Han, S.-D.; Pan, J.; Li, J.-H.; Wang, G.-M.; Wang, Z.-H. *Cryst. Growth Des.* **2017**, *17*, 3588–3591.
- (805) Zhang, M.; Sheng, T.; Wang, X.; Hu, S.; Fu, R.; Chen, J.; He, Y.; Qin, Z.; Shen, C.; Wu, X. *CrystEngComm* **2009**, *12*, 73–76.
- (806) Shmilovits, M.; Diskin-Posner, Y.; Vinodu, M.; Goldberg, I. *Cryst. Growth Des.* **2003**, *3*, 855–863.

- (807) Garin, A. B.; Rakarić, D.; Andrić, E. K.; Kosanović, M. M.; Balić, T.; Perdih, F. *Polyhedron* **2019**, *166*, 226–232.
- (808) Zi, G.; Wang, Q.; Xiang, L.; Song, H. *Dalton Trans.* **2008**, 5930–5944.
- (809) Liao, P.-K.; Shi, D.-R.; Liao, J.-H.; Liu, C. W.; Artem'ev, A. V.; Kuimov, V. A.; Gusarova, N. K.; Trofimov, B. A. *Eur. J. Inorg. Chem.* **2012**, *2012*, 4921–4929.
- (810) Yuan, M.; Gao, S.; Zhao, F.; Zhang, W.; Wang, Z. *Sci. China Ser. B-Chem.* **2009**, *52*, 266–275.
- (811) Belo, D.; Figueira, M. J.; Mendonça, J.; Santos, I. C.; Almeida, M.; Henriques, R. T.; Duarte, M. T.; Rovira, C.; Veciana, J. *Eur. J. Inorg. Chem.* **2005**, *2005*, 3337–3345.
- (812) Yang, L.; Powell, D. R.; Houser, R. P. *Dalton Trans.* **2007**, 955–964.
- (813) Dryden, N. H.; Puddephatt, R. J.; Roy, S.; Vittal, J. J. *Acta Cryst. C* **1994**, *50*, 533–536.
- (814) Degnan, I. A.; Behm, J.; Cook, M. R.; Herrmann, W. A. *Inorg. Chem.* **1991**, *30*, 2165–2170.
- (815) Pérez-Cordero, E. E.; Campana, C.; Echegoyen, L. *Angew. Chem. Int. Ed.* **1997**, *36*, 137–140.
- (816) England, J.; Scarborough, C. C.; Weyhermüller, T.; Sproules, S.; Wieghardt, K. *Eur. J. Inorg. Chem.* **2012**, *2012*, 4605–4621.
- (817) Monteiro, J.; Ros, J.; Skylogiani, E.; Hartono, A.; Svendsen, H.; Knuutila, H.; Moser, P.; Wiechers, G.; Charalambous, C.; Garcia, S. *Accelerating Low carbon Industrial Growth through CCUS Deliverable Nr. D1.1.7 Guidelines for Emissions Control*; tech. rep.; 2021, p 43.
- (818) Mechleri, E.; Lawal, A.; Ramos, A.; Davison, J.; Dowell, N. M. *Int. J. Greenh. Gas Con.* **2017**, *57*, 14–25.
- (819) Bui, M.; Gunawan, I.; Verheyen, V.; Feron, P.; Meuleman, E. *Int. J. Greenh. Gas Con.* **2016**, *48*, 188–203.
- (820) Bai, S.; Kolter, J. Z.; Koltun, V. In *arXiv preprint Arxiv-1803.01271*, 2018.
- (821) He, K.; Zhang, X.; Ren, S.; Sun, J. *CVPR* **2016**.
- (822) Zhu, L.; Laptev, N. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2017, pp 103–110.
- (823) Hron, J.; de G. Matthews, A. G.; Ghahramani, Z. In *arXiv preprint Arxiv-1711.02989*, 2017.
- (824) Hron, J.; de G. Matthews, A. G.; Ghahramani, Z. In *arXiv preprint Arxiv-1807.01969*, 2018.
- (825) Lim, B.; Arik, S. O.; Loeff, N.; Pfister, T. In *arXiv preprint Arxiv-1912.09363*, 2019.
- (826) Eichler, M. *J. Econometrics* **2007**, *137*, 334–353.
- (827) Crespi, S.; Simeth, N. A.; König, B. *Nat. Rev. Chem.* **2019**, *3*, 133–146.
- (828) Savjani, K. T.; Gajjar, A. K.; Savjani, J. K. *ISRN Pharm.* **2012**, *2012*, 1–10.
- (829) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. *J. Comput. Aided Mol. Des.* **2016**, *30*, 595–608.
- (830) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. In *Advances in Neural Information Processing Systems*, ed. by Cortes, C.; Lawrence, N.; Lee, D.; Sugiyama, M.; Garnett, R., Curran Associates, Inc.: 2015; Vol. 28.

- (831) Waring, M. J. *Expert Opin. Drug Discovery* **2010**, *5*, 235–248.
- (832) Yaghi, O. M. *ACS Cent. Sci.* **2019**, *5*, 1295–1300.
- (833) Burtch, N. C.; Jasuja, H.; Walton, K. S. *Chem. Rev.* **2014**, *114*, 10575–10612.
- (834) Yeh, J.-W.; Chen, S.-K.; Lin, S.-J.; Gan, J.-Y.; Chin, T.-S.; Shun, T.-T.; Tsau, C.-H.; Chang, S.-Y. *Adv. Eng. Mater.* **2004**, *6*, 299–303.
- (835) Cantor, B.; Chang, I.; Knight, P.; Vincent, A. *Materials Science and Engineering: A* **2004**, 375–377, 213–218.
- (836) George, E. P.; Raabe, D.; Ritchie, R. O. *Nat. Rev. Mater.* **2019**, *4*, 515–534.
- (837) Pomberger, A.; McCarthy, A. A. P.; Khan, A.; Sung, S.; Taylor, C. J.; Gaunt, M. J.; Colwell, L.; Walz, D.; Lapkin, A. A. *React. Chem. Eng.* **2022**, *7*, 1368–1379.
- (838) Hickman, R.; Ruža, J.; Roch, L.; Tribukait, H.; Garcia-Durán, A. **2022**.
- (839) Lu, J.; Xia, S.; Lu, J.; Zhang, Y. *J. Chem. Inf. Model.* **2021**, *61*, 1095–1104.
- (840) Ho, J.; Tumkaya, T.; Aryal, S.; Choi, H.; Claridge-Chang, A. *Nat. Methods* **2019**, *16*, 565–566.
- (841) Lester, B.; Al-Rfou, R.; Constant, N. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, 7–11 November, 2021*, ed. by Moens, M.-F.; Huang, X.; Specia, L.; Yih, S. W.-t., Association for Computational Linguistics: 2021, pp 3045–3059.
- (842) Singhal, K. et al. *arXiv preprint arXiv: Arxiv-2212.13138* **2022**.
- (843) Griffiths, R.-R.; Klarner, L.; Moss, H.; Ravuri, A.; Truong, S. T.; Rankovic, B.; Du, Y.; Jamasb, A. R.; Schwartz, J.; Tripp, A., et al. In *ICML 2022 2nd AI for Science Workshop*.
- (844) Probst, D.; Schwaller, P.; Reymond, J.-L. *Digital Discovery* **2022**, *1*, 91–97.
- (845) Schwaller, P.; Probst, D.; Vaucher, A. C.; Nair, V. H.; Kreutter, D.; Laino, T.; Reymond, J.-L. *Nat. Mach. Intell.* **2021**, *3*, 144–152.
- (846) Faber, F.; Lindmaa, A.; von Lilienfeld, O. A.; Armiento, R. *Int. J. Quantum Chem.* **2015**, *115*, 1094–1101.
- (847) Breiman, L. *Mach. Learn.* **2001**, *45*, 5–32.
- (848) Eriksson, D.; Jankowiak, M. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, 27–30 July 2021*, ed. by de Campos, C. P.; Maathuis, M. H.; Quaeghebeur, E., AUAI Press: 2021; Vol. 161, pp 493–503.
- (849) Frisch, M. J. et al. Gaussian16 Revision C.01, Gaussian Inc. Wallingford CT, 2016.
- (850) Jacquemin, D.; Preat, J.; Perpète, E. A.; Vercauteren, D. P.; André, J.-M.; Ciofini, I.; Adamo, C. *Int. J. Quantum Chem.* **2011**, *111*, 4224–4240.
- (851) Adamo, C.; Barone, V. *J. Chem. Phys.* **1999**, *110*, 6158–6170.
- (852) Petersson, G. A.; Bennett, A.; Tensfeldt, T. G.; Al-Laham, M. A.; Shirley, W. A.; Mantzaris, J. *J. Chem. Phys.* **1988**, *89*, 2193–2218.
- (853) Petersson, G. A.; Al-Laham, M. A. *J. Chem. Phys.* **1991**, *94*, 6081–6090.
- (854) Tomasi, J.; Mennucci, B.; Cammi, R. *Chem. Rev.* **2005**, *105*, 2999–3094.
- (855) Jablonka, K. M. givemeconformer, <https://github.com/kjappelbaum/givemeconformer>, accessed 2023-3-9, 2022.
- (856) Halgren, T. A. *J. Comput. Chem.* **1996**, *17*, 490–519.

- (857) Hyland, B.; Atemez, G.; Pendleton, M.; Srivastava, B. Linked Data Glossary, accessed 2021-1-8, 2013.
- (858) Uschold, M. *Appl. Onto.* **2015**, *10*, 243–258.
- (859) Degtyarenko, K.; de Matos, P.; Ennis, M.; Hastings, J.; Zbinden, M.; McNaught, A.; Alcantara, R.; Darsow, M.; Guedj, M.; Ashburner, M. *Nucleic Acids Res.* **2007**, *36*, D344–D350.
- (860) Pachl, C.; Frank, N.; Breitbart, J.; Bräse, S. In *arXiv preprint Arxiv-2002.03842*, 2020.
- (861) Chalk, S. The IUPAC Gold Book: Toward a Chemical Ontology for the Next 100 Years of Chemistry, <https://iupac.org/100/stories/the-iupac-gold-book/>, accessed 2023-3-9, 2019.
- (862) Gipp, B.; Meuschke, N.; Gernandt, A. In *Proceedings of the iConference 2015*, Newport Beach, CA, USA, 2015.
- (863) Fridman, L. Brendan Eich: JavaScript, Firefox, Mozilla, and Brave | Lex Fridman Podcast #160, <https://www.youtube.com/watch?v=krBOenBeSiE>, accessed 2023-3-9, 2021.

GLOSSARY

D	set of discarded points.
E	design space.
P	(ϵ -accurate) Pareto set.
$Q_{\mu,\sigma,\beta}(\mathbf{x})$	uncertainty hyperrectangle of point \mathbf{x} .
R_g	radius of gyration.
$R_t(x)$	uncertainty region of point \mathbf{x} .
U	set of unclassified points.
ΔG_{ads}	free energy of adsorption.
ΔG_{rep}	repulsive free energy of polymer dimer.
β_t	scaling parameter for the hyperrectangle.
access token	are random character strings that can, similar to passwords, be used to grant the rights for certain operations. Usually, access tokens can be minted for a limited scope, e.g., for only one sample and only specific rights (e.g., read rights).
AMD	average minimum distance.
AMP	2-amino-2-methyl-1-propanol.
API	an Application Programming Interface (API) describes how one can interact with a program. It describes what requests can be made and in which form they must be made. In contrast to a user interface it is not intended for direct use via a “frontend” but rather to offer programmatic access.
APRDF	atomic-property labeled radial distribution function.
ARD	automatic relevance determination.
AUC	area under the ROC curve.
AUPR	area under the precision-recall curve.
backend	the piece of software that accesses the data. One standard model in software development is the client-server model, where the backend is the part of the code that runs on the server.
BO	Bayesian optimization.
BV	bond valence.
CIF	Crystallographic Information File.
client	the archetypal example of a client is a web browser. A client can be used to access a service offered by a server.
CNN	convolutional neural network.
COD	Crystallographic Open Database.
COF	covalent-organic framework.
CoRE	Computation-Ready, Experimental.
CrabNet	compositionally-restricted attention-based network.
CSD	Cambridge Structural Database.
CV	collective variable.

DCC	direct contact cooler.
DDEC	density-derived electrostatic charge.
deployment	deployment refers to the activities that are needed to make a piece of software available to users.
DFT	density functional theory.
disruptive innovation	a disruptive innovation is an innovation that creates a new market. One example of this are smartphones.
DoE	design of experiments.
DOI	a digital object identifier (DOI) is a <i>unique and persistent</i> identifier that is created by some central agency that keeps track of some metadata and commits to resolving it indefinitely.
DPD	dissipative particle dynamics.
Dublin Core	the Dublin Core Metadata Element Set is a set of 15 elements like “Contributor”, “Date”, “Format” that should be used for describing data resources. A similar set of metadata is the DataCite Metadata Schema.
ECFP	extended connectivity fingerprint.
EDA	exploratory data analysis.
EFG	extended functional group.
EGO	efficient global optimization.
EI	expected improvement.
ELN	in this work, we see an electronic lab notebook (ELN) not only as a place where experimental chemists take notes but as a hub for chemical research. Note-taking is an important functionality in this hub, but visualizing, interacting, and sharing data are at least equally important.
EPR	electron paramagnetic resonance.
et	extra trees.
FAIR	the FAIR principles are guidelines on how the findability, accessibility, interoperability, and reusability of data can be improved. ⁴² Importantly, FAIR does not necessarily mean open.
fcc	face-centered cubic.
five star data	Tim Berners-Lee suggests that the highest-quality open and linked data, i.e. 5-star data, ⁸⁵⁷ must (1) be available on the web under an open license (e.g., creative commons), (2) be structured (e.g., table instead of the image of a table), (3) be non-proprietary (e.g., using standard text files instead of binary formats), (4) use URIs (e.g, using an URL such that others can share and link to this piece of data), (5) link to other data. In contrast to FAIR, linked open data (5-star data) must be open.
FN	false negative.
FP	false positive.

frontend	the piece of software that displays the data, it is the part of a website the users interact with. In the client-server model this is the piece of software that runs on the client (e.g., the web browser).
GA	genetic algorithm.
GAN	generative adversarial neural network.
gb	gradient boosting.
GBDT	gradient boosted decision tree.
GCMC	grand canonical Monte Carlo.
GGA	generalized gradient approximation.
GitHub	github.com is where most of the open source code “lives” and is collaboratively built. It is integrated with the version control tool Git but also gives the option to make copies of other code bases (“fork”/“clone”) or to contribute code to some projects using “pull requests”. Bugs and new features are usually discussed in “issues”.
GNN	graph neural network.
GPR	Gaussian process regression.
GPT-3	generative pre-trained transformer model 3.
HEA	high entropy alloy.
HEX	heat exchanger.
HOMO	highest occupied molecular orbital.
HTVS	high-throughput virtual screening.
ICM	intrinsic coregionalization model.
ICSD	Inorganic Crystal Structure Database.
IUPAC	International Union of Pure and Applied Chemistry.
JSON	JavaScript object notation.
Kaggle	is the largest community for data science and most known for its competitions in which participants compete to create the best model for a task posed by the host of the competition.
knn	k-nearest neighbors.
L-BFGS	limited memory Broyden–Fletcher–Goldfarb–Shanno.
LAMMPS	Large-scale Atomic/Molecular Massively Parallel Simulator.
LCA	life-cycle analysis.
LDR	labeled data record.
LIMS	laboratory infrastructure management systems allow to track and manage samples. Often they allow defining locations at which different material or equipment is stored and can then track the moves and uses of the materials.
link rot	link rot describes the phenomenon that links tend to cease to resolve to the content they originally pointed to.
LLM	large language model.

LQG	labeled quotient graph.
LUMO	lowest unoccupied molecular orbital.
MAE	mean absolute error.
markup	a markup language allows annotating the text to decouple the structure from the content. One example is \LaTeX where the <code>\section</code> command can be used to define the concept of a section. In the context of data, markup languages such as Extensible Markup Language (XML) are typically used. The archetypal example for chemistry is CML.
MCC	Matthews correlation coefficient.
MD	molecular dynamics.
MEA	monoethanolamine.
metadata	metadata is data that describes attributes of other data and in this way provides context and makes data discoverable. While this might sound abstract there are many use cases in which metadata can be helpful. For example, date and time should be featured in most metadata records. If we have access to them we can use them to filter our data, but also correlate them with other information (such as the weather reports) to understand potential outliers in our data. A useful overview of metadata schemes is https://rdamsc.bath.ac.uk/ .
MHFP	MinHash fingerprint.
ML	machine learning.
MMD	maximum mean discrepancy.
MOF	metal-organic framework.
MPNN	message-passing NN.
NLP	natural language processing.
NMR	nuclear magnetic resonance.
NN	neural network.
NNGP	neural network Gaussian process.
NTK	neural tangent kernel.

ontology	an ontology describes and clarifies the meaning, conceptualization, and relation of terms in a formal language. ⁸⁵⁸ That is, they go beyond the simple vocabulary or thesaurus by also providing a model for the information. Ontologies are often written in the web ontology language (OWL). For example in the ChEBI ontology ⁸⁵⁹ “metal atom” is defined as “An atom of an element that exhibits typical metallic properties, being typically shiny, with high electrical and thermal conductivity” and the relation to other terms is formally described (i.e., on could make mathematical inferences, e.g., on the consistency). In the context of chemical data and ELNs ontologies are important to make sure that it is clear and well-defined what certain items in a database mean (for example, the concept “bond order” depends on the definition—for this reason, a “bond order” entry in a database would link to an ontology with the definition). Importantly, ontologies are relevant to enable semantic interoperability, i.e., enabling the decoding of the meaning of data. Pachl et al. ⁸⁶⁰ give an excellent overview of ontologies that have been developed for chemistry (and find that many projects remain unfinished). One example that provides a controlled vocabulary that can be programmatically accessed using an API is the IUPAC gold book. However, since a vocabulary is not a full ontology, there are ongoing efforts also to model the relationship between terms to arrive at a full chemical ontology. ⁸⁶¹
open notebook science	popularized by Jean-Claude Bradley, ⁹³ open notebook science refers to making <i>all</i> primary research outputs openly available as they are recorded. The goal is to minimize the amount of “dark data”, e.g., “unsuccessful” experiments that would never get published, and “insider information” that one might not report in methods sections in papers. openlabnotebooks.org collects some (mostly life-science) efforts.
open source	open source software is software that is openly available for modification and reuse. Development of open source software is often decentralized, and happens in open collaboration.
OPV	organic photovoltaics.
<i>Organic Syntheses</i>	<i>Organic Syntheses</i> is a journal that publishes procedures for the synthesis of organic compounds. The key distinction is that the syntheses are replicated in a lab of one of the members of the editorial board.
PAL	Pareto active learning.
PBE	Perdew-Burke-Ernzerhof.
PC	principal component.
PCA	principal component analysis.
PCC	post-combustion capture.
PCE	power conversion efficiency.
PCM	Polarizable Continuum Model.

PFN	Prior-Data Fitted Network.
PMF	potential of mean force.
PPD	posterior predictive distribution.
proprietary	proprietary files contain data that are stored in an encoding scheme defined by an organization. The difficulty with proprietary files stems from the fact that the schema is secret, that is, it is often not trivial to decode and interpret the files. For this reason, data can only be five star data if it is shared in a non-proprietary format.
provenance	data provenance is a form of metadata that describes the origin of data.
PSO	particle swarm optimization.
pXRD	powder X-ray diffraction.
Pz	piperazine.
RAC	revised autocorrelation function.
RDF	Resource Description Framework (RDF) triplestores break down information into triples of “subject”, “verb”, “object” that form a labeled graph which machines can use to explore information (and understand how information is connected). An example of a triple can be “substance” (subject) “inhibits” (verb) “protein” (object) and the same element can also be used in “publication” (subject) “describes” (verb) “protein”.
repository	is a location where software (in the case of software repositories, e.g., on GitHub) or data are stored/archived (e.g., Zenodo). www.re3data.org/ is a useful resource to find suitable repositories.
REST	(Representational State Transfer (REST)). One key design principle of RESTful interfaces is that they are stateless. Practically speaking this means that the clients will send <i>all information</i> that is needed to process a request to a <i>server</i> . That is, the client, and not the server, stores all the information about the client state. Importantly, the REST scheme also defines a uniform structure for the APIs.
RF	random forest.
RKHS	reproducing kernel Hilbert space.
ROC	receiver-operating characteristic.
schema	a schema describes the structure of the data in a formal language. JSON-LD and XML are common formats for schemas. Examples in the context of the sciences are XML schema of the Chemical Markup Language (CML) ⁴⁶² or the Analytical Information Markup Language (AnIML) ⁴⁵⁷ .
SEFLIES	self-referencing embedded strings.
server	is the computer (program) that provides some functionality for the clients. A typical example is a file server that provides files to its client (as it is for example done in companies). One typical server that is used in ELNs is the database server that serves the database to the clients.

sgd	stochastic gradient descent.
SHAP	SHapley Additive exPlanations.
SLA	service level agreement.
SMILES	simplified molecular-input line-entry system.
SOAP	smooth overlap of atomic positions.
TabPFN	tabular PFN.
TDDFT	time-dependent DFT.
TMAP	tree-map.
TN	true negative.
TP	true positive.
trusted timestamps	trusted timestamps proof that certain information existed at a certain point in time. Importantly, they are secure in the sense that also the owner of the document cannot change the timestamp. They can be secured by techniques like the blockchain ⁸⁶² or a time-stamping authority.
UMAP	uniform manifold approximation and projection for dimension reduction.
UQG	unlabeled quotient graph.
URI	Uniform Resource Identifiers (URIs, as defined in RFC 3986) are sequences of characters that identify a physical (e.g., sample) or digital (e.g., web page) resource. One special example are Uniform Resource Locators (URLs). Internationalized Resource Identifiers (IRIs) are generalizations of URIs to Unicode glyphs. Note that there are initiatives, such as the Resource Identification Portal, that provide a centralized location for URIs for research resources such as antibodies or cell lines.
VAE	variational autoencoder.
VSEPR	valence-shell electron-pair repulsion.
web browser	in an abstract sense, a web browser is a piece of software that can be used to retrieve information from the world wide web. Practically speaking it can “speak” certain programming languages like JavaScript and HTML. This means that a program that is written in JavaScript can be executed by all major browsers on any platform or any device. This led some people to say that a web browser is a “super app” ⁸⁶³ that could potentially replace almost all other apps on a computer. This is important in the context of ELNs as this might allow us to create systems that are easier to scale (one does not have to develop a separate app for every operating system).
WHAM	weighted histogram analysis method.
XANES	X-ray absorption near edge structure.
XPS	X-ray photoelectron spectroscopy.
xTB	extended tight-binding.

KEVIN MAIK JABLONKA

- 🏠 EPFL Valais Wallis, Rue de l'Industrie 17, 1951 Sion, Switzerland
- ✉ mail@kjablonka.com
- 🔗 kjappelbaum (among the 100 most active users in Switzerland)
- 🎓 Google Scholar ID R2ntl8IAAAAJ

EDUCATION

Ph.D. in chemistry and chemical engineering Aug 2019 – Apr 2023

École polytechnique fédérale de Lausanne (EPFL), Switzerland.

- Thesis: *Toward data-driven materials design: From atoms to pilot plants* under supervision of Prof. Berend Smit.

MSc. in chemistry (high distinction, 5.95/6.00) Sep 2017 – July 2019

École polytechnique fédérale de Lausanne (EPFL), Switzerland.

- Thesis: *Expediting the Discovery of High-Performance Porous Materials*. Analyzed the applicability of tail corrections for grand-canonical Monte-Carlo simulations and used neural networks to predict gas adsorption based on diffraction patterns. Under the supervision of Prof. Berend Smit.
- Research internship in the ultrafast spectroscopy group of Prof. Majed Chergui on the cooling of excited states in ZnO, including beamtime at the Swiss Light Source.
- 2nd best GPA among all master graduates at EPFL, best GPA in chemistry.

Certificate of Extended Studies in Applied Data Science Feb 2019

École polytechnique fédérale de Lausanne (EPFL), Switzerland.

BSc. in chemistry (high distinction, 1.2/1.0) Oct 2014 – Aug 2017

Technical University of Munich, Germany.

- Thesis: *Towards Studies on Metal-Cluster/Semiconductor-Hybridmaterials for Photocatalysis*. Deposition of size-selected clusters in ultrahigh vacuum, contributed to the re-building of the setup and developed a toolkit for analyzing the cluster size distributions). Under the supervision of Prof. Ueli Heiz.

Abitur (1.1/1.0) Sep 2006 – Jul 2014

Wieland Gymnasium, Biberach an der Riss, Germany.

WORK EXPERIENCE

Intern Quantum Chemistry & Molecular Simulation Sep 2018 – Feb 2019

BASF Corporation, Tarrytown, NY, USA.

- Implemented a workflow for training ML models to predict the thermodynamic stability of zeolites (including automatic structure generation and DFT calculations) that is now regularly used by researchers at BASF.
- Built ML models to predict surfactant properties which led to a collaboration on using active learning for Pareto-optimal materials design.

Scientific Support Non-Clinical Pharmacokinetics Jul 2014 – Sep 2014

Boehringer-Ingelheim, Biberach an der Riss, Germany.

- Analysis of drug adsorption to melanin with radioactive markers and developed protocol for analysis of lysosomal trapping with Raman microscopy which led to a new collaboration within Boehringer-Ingelheim.

SOFTWARE DEVELOPMENT

- Python (daily), JavaScript/TypeScript (weekly), Julia, HTML, CSS, Bash, C.
- Using Git, Docker, and test-driven development on a daily basis.
- Developed toolkits for machine learning for reticular chemistry, mofdscribe.
- Contributor in the mljs, cheminfo, and cheminfo-py organizations.
- Developed electronic lab notebook (ELN) plugins, among others, for X-ray diffraction, gas adsorption isotherms, thermogravimetric analysis, baseline correction, X-ray photoelectron spectroscopy, PubChem API access and safety information compilation.

HONORS AND AWARDS

- Outstanding reviewer for Digital Discovery (2022).
- Swiss Academy of Sciences Chemistry Travel Award (1 kCHF).
- Member of the CAS FutureLeaders 2022 cohort.
- 2nd best GPA among all master graduates at EPFL, best GPA in chemistry (BASF prize, 2019).
- Alfred-Werner scholarship (approx. 20 kCHF) by the Swiss Chemical Society (2017).
- Member of the BASF European Talent Pool (2015–).
- Scholarship by the German scholarship foundation (Studienstiftung des Deutschen Volkes, 2015).
- Scholarship Deutschlandstipendium (2015).
- 3rd prize in Chemistry and *future technology prize* (awarded by the Federal Ministry for Education and Research) at the national German science fair “Jugend forscht” (2014).
- Anton-von-Störck prize (and badge of honor of town Bad Saulgau) for achievements in science (2015).
- Finalist at the Siemens science fair (2014).
- Silver medal at the International Conference of Young Scientists in Bali, Indonesia (2013).
- Prize for the best graduation in chemistry awarded by the German chemical industry association (2014).
- Prize of the Chamber of Industry and Commerce for the best graduation in science (2014).
- Hilde-Frey Prize for the best graduation in the county and Hilde-Frey special award for extracurricular commitment in the county (Landkreis Biberach, 2014).

TEACHING AND SUPERVISION

- Supervisions of multiple (>5) master projects and internships.
- Lecture and hands-on workshop on machine learning for the MolSim winter school (University of Amsterdam) (2020, 2021, 2022, 2023). Designed flipped classroom version with quizzes using the Moodle and Discord platforms.
- Guest lecture on research data management in chemistry in a course on research data management at the EPFL doctoral school (2021, 2022, 2023).
- Organizer of and lecturer in the doctoral course on using the EPFL electronic lab notebook (1 ECTS, spring 2021, fall 2022).
- Lectures and exercises on computational carpentry, data in chemistry/open science, and machine learning (2020, 2021, 2022). Designed course material and exercises for the computational carpentry, data in chemistry, and machine learning parts and delivered those lectures.
- Teaching assistant for undergraduate chemistry courses “principles and methods of chemistry” (winter 2015), “physical chemistry 1” (winter 2015 and summer 2016) and “introduction to quantum mechanics” (winter 2016) at Technical University of Munich.

FUNDING

- Co-lead of ChemNLP project in the OpenBioML collaborative research laboratory supported by stability.ai.
- PI on Digital Resources for Instruction and Learning fund (20 kCHF) for the development of virtual spectroscopy laboratory.
- Co-author on proposal for interfaculty funding “Graph Learning for Reticular Chemistry” (120 kCHF) jointly between the PIs Berend Smit and Sofia Olhede (math department at EPFL).
- Co-author on EPFL platform seed fund “Supporting the analysis of XPS data by quantum chemistry: making simulated photoelectron spectra routine” (90 kCHF) jointly between the PIs Berend Smit, Nicola Marzari, and the X-ray spectroscopy platform.
- Co-investigator on CSCS projects “Computational Screening of Covalent- and Metal-Organic Frameworks for Applications in Photocatalysis” (510 kCPUh) and “Thermal Conductivity of MOFs” (200 kCPUh).
- Co-investigator on PRACE project “Thermal conductivity of Metal-Organic-Frameworks” (80 10⁶ CPUh).
- Co-investigator on BIG-MAP stakeholder initiative project “Interoperable data management for fundamental battery research” (150 k). Initiated the collaboration with the University of Cambridge, UC Louvain, Zakodium Sàrl, and EPFL.

ACADEMIC SERVICE

- Co-organizer of “March madness” hackathon on Large-Language Models in materials science (more than 200 registrations, 2023)
- Program committee/reviewer Chemistry & Materials Science and Machine Learning & Data Science for SciPy 2022
- Lead-Organizer and initiator of the CECAM workshop on Machine-Actionable Data Interoperability for Chemical Sciences 2022.
- Letters to a Pre-Scientist (giving students in low-income classrooms insights into the life of a STEM professional via snail mail, 2021).
- Co-organizer of the seminar series “Artificial intelligence in chemistry and beyond” at EPFL (also offered as doctoral course, 2021–).
- Reviewer for the *Proceedings of the National Academy of Sciences*, *npj computational materials*, *JACS Au*, *Scientific Data*, *Chemical Science*, *ICLR*, *Journal of Chemical Information and Modeling*, *Journal of Chemical Theory and Computation*, *Johnson Matthey Technology Review*, *Digital Discovery*, *Journal of Materials Science A*, *Journal of Open Source Software*, *ACS Omega*
- Member of the “Trainee advisory committee” of the *Living Journal of Computational Molecular Science*.
- “Data Champion” at EPFL, supporting the EPFL community in research data management and data analysis questions (2020–).
- Juror at the German national student science fair “Jugend forscht” (2020, 2022, 2023).
- Supervisor of electrospinning projects at a Student Research Center as well as supervisor of the gas chromatography course and mentor of the laboratory course (2017–2020).

INVITED TALKS/SEMINARS

- Lab AI team at Solvay (Apr 2023)
- Center for Energy and Environmental Chemistry Jena (Jan 2023)
- WWU and Helmholtz-Institut Münster, Seminar series “Machine Learning in Physical Chemistry – 2022–23” (Jan 2023)
- NFDI4Chem Stammtisch (May 2022)
- University of Cambridge, Lapkin group meeting (February 2022)
- ETH Zurich, Mini-Symposium on Digital Chemistry (November 2021)
- IBM Research Zurich (November 2021)
- NCCR MARVEL Junior seminar (December 2021)
- EPFL, Nicola Marzari group meeting (December 2021)

KEVIN MAIK JABLONKA

PUBLICATIONS

As first author

1. Kevin Maik Jablonka, Andrew S. Rosen, Aditi S. Krishnapriyan, Berend Smit, "An ecosystem for digital reticular chemistry", *ACS Central Science*, **2023**.
Highlighted as In Focus article. and news outlets such as Chemistry World.
2. Kevin Maik Jablonka, Charithea Charalambous, Georg Wiechers, Eva Sanchez Fernandez, Juliana Monteiro, Peter Moser, Berend Smit, Susana Garcia, "Machine learning for industrial processes: Forecasting amine emissions from a carbon capture plant", *Science Advances* **2023**, 9, eadc9576.
See also the feature on the EPFL landing page. Highlighted in Carbon Capture Journal Jan/Feb 2023.
3. Kevin Maik Jablonka, Luc Patiny, Berend Smit, "Making the collective knowledge of chemistry open and machine actionable", *Nature Chemistry*, **2022** 14, 365–376.
Featured, among others, on Phys.org.
4. Kevin Maik Jablonka, Luc Patiny, Berend Smit: "Making molecules vibrate: Interactive web environment for the teaching of infrared spectroscopy", *Journal of Chemical Education*, **2022** 99, 2, 561–569.
See also the feature on the EPFL landing page.
5. Kevin Maik Jablonka, Daniele Ongari, Seyed Mohamad Moosavi, and Berend Smit, "Using collective knowledge to assign oxidation states of metal cations in metalorganic frameworks", *Nature Chemistry*, **2021** 13, 771–777.
Featured, among others, in Nature Review Materials and Materials Today.
6. Kevin Maik Jablonka, Giriprasad Melpatti Jothiappan, Shefang Wang, Berend Smit, Brian Yoo, "Bias free multiobjective active learning for materials design and discovery", *Nature Communications*, **2021** 12, 2312.
Part of the collection "Computation and Machine Learning for Chemistry" and "AI and machine learning". Picked for "Reading list AI in materials discovery" in C&EN Discovery report 2021.
7. Kevin Maik Jablonka, Seyed Mohamad Moosavi, Mehrdad Asgari, Christopher Ireland, Luc Patiny, and Berend Smit, "A Data-Driven Perspective on the Colours of Metal-Organic Frameworks", *Chemical Science*, **2020** 12, 3587–3598.
Featured in Chemistry World News, "Swiss science concentrates" in CHIMIA, and "Notizen aus der Chemie".
8. Kevin Maik Jablonka, Daniele Ongari, Seyed Mohamad Moosavi, and Berend Smit, "Big-Data Science in Porous Materials: Materials Genomics and Machine Learning", *Chemical Reviews*, **2020** 120, 8066–8129.
9. Kevin Maik Jablonka, Daniele Ongari, Berend Smit, "Applicability of Tail Corrections in the Molecular Simulations of Porous Materials". *Journal of Chemical Theory and Computation* **2019**, 15, 5635–5641.

As coauthor

10. Nency P. Domingues, Seyed Mohamad Moosavi, Leopold Talirz, Kevin Maik Jablonka, Christopher P. Ireland, Fatmah Mish Ebrahim, Berend Smit, "Using genetic algorithms to systematically improve the synthesis conditions of Al-PMOF", *Communications Chemistry*, **2022**, 5, 170.
11. Mario Krenn, Qianxiang Ai, Senja Barthel, Nessa Carson, Angelo Frei, Nathan C. Frey, Pascal Friederich, Théophile Gaudin, Alberto Alexander Gayle, Kevin Maik Jablonka, Rafael F. Lameir, Dominik Lemm, Alston Lo, Seyed Mohamad Moosavi, José Manuel Nápoles-Duarte, AkshatKumar Nigam, Robert Pollice, Kohulan Rajan, Ulrich Schatzschneider, Philippe Schwaller, Marta Skreta, Berend Smit, Felix Strieth-Kalthoff, Chong Sun, Gary Tom, Guido Falk von Rudorff, Andrew Wang, Andrew White, Adamo Young, Rose Yu, and Alán Aspuru-Guzik, "SELFIES and the future of molecular string representations", *Patterns*, **2022**, 3, 100588.

12. Sterling G. Baird, Kevin M. Jablonka, Michael D. Alverson, Hasan M. Sayeed, Mohammed Faris Khan, Colton Seegmiller, Berend Smit, Taylor D. Sparks, "xtal2png: A Python package for representing crystal structure as PNG files", *Journal of Open Source Software*, **2022**, 7, 4528.
13. Alicia Lund, Manohara Gudiya, Ah-Young Song, Kevin Maik Jablonka, Christopher Ireland, Li Anne Cheah, Berend Smit, Susana Garcia, Jeffrey Reimer, "Characterization of Chemisorbed Species and Active Adsorption Sites in Mg-Al Mixed Metal Oxides for High Temperature CO₂ Capture", *Chemistry of Materials*, **2022**, 34, 3893–3901.
14. Daniele Ongari, Leopold Talirz, Kevin Maik Jablonka, Daniel W. Siderius, Berend Smit, "Data-driven matching of crystals and experimental isotherms of Metal-Organic Frameworks", *Journal of Chemical & Engineering Data*, **2022**, 67, 1743–1756.
15. Sauradeep Majumdar, Seyed Mohamad Moosavi, Kevin Maik Jablonka, Daniele Ongari, Berend Smit, "Diversifying databases of metal-organic frameworks for high-throughput computational screening", *ACS Applied Materials and Interfaces* **2021**, 13, 61004–61014.
16. Seyed Mohamad Moosavi, Aditya Nandy, Kevin Maik Jablonka, Daniele Ongari, Jon Paul Janet, Peter G. Boyd, Yongjin Lee, Berend Smit, and Heather J. Kulik, "Understanding the diversity of the metal-organic frameworks ecosystem", *Nature Communications* **2020**, 11, 1, 1–10.
17. Seyed Mohamad Moosavi, Kevin Maik Jablonka, and Berend Smit, "The Role of Machine Learning in the Understanding and Design of Materials", *Journal of the American Chemical Society* **2020**, 142, 20273–20287.
18. Maria Fumanal, Andres Ortega-Guerrero, Kevin Maik Jablonka, Berend Smit, Ivano Tavernelli, "Charge Separation and Charge Carrier Mobility in Photocatalytically Active Metal-Organic Frameworks", *Advanced Functional Materials* **2020**, 30, 2003792.

Preprint & Unpublished

19. Jacky Deng; Salma Ahmed, Ernest, Awoonor-Williams, Proгна Banerjee, Magda Barecka, Laura Bickerton, Silvina Di Pietro, Stanna Dorn, Kevin Jablonka; Gabriele Laudadio, Elisabeth Kreidt, Helena Mannochio-Russo, Júlio Terra, Olivia Wilkins, Saigopalakrishna Yerneni, Maha Yusuf, "Prioritizing Mentorship as Scientific Leaders", submitted, **2023**.
20. Beatriz Mourino, Andres Ortega-Guerrero, Kevin Maik Jablonka, and Berend Smit "DFT-based discovery of experimental covalent organic frameworks for photocatalysis". Available on ChemRxiv: 10.26434/chemrxiv-2023-515dc-v2.
21. Kevin Maik Jablonka, Philippe Schwaller, Andres Ortega-Guerrero, Berend Smit, "Is GPT-3 all you need for low-data discovery in chemistry?", NeurIPS AI4Mat and ELLIS ML4Molecules workshop **2022**. Available on ChemRxiv 10.26434/chemrxiv-2023-fw8n4. *Highlighted in MIT Technology Review*.
22. Fatmah Mish Ebrahim, Maria Fumanal, Andrzej Gadysiak, Özge Kadioglu, Kevin Maik Jablonka, Daniele Ongari, Amber Mace, Serhii Shyshkanov, Sergio Saris, Christopher Patrick Ireland, Paul J. Dyson, Kyr- iakos C. Stylianou, Berend Smit, "Tuneable luminescence from a biomolecule-inspired single-species emitter of white light", submitted, **2021**. Available on ChemRxiv: 10.33774/chemrxiv-2021-dg13d.
23. Kevin Maik Jablonka, Michaël Zasso, Luc Patiny, Nicola Marzari, Giovanni Pizzi, Berend Smit, Aliak- sandr V. Yakutovich, "Connecting lab experiments with computer experiments: Making 'routine' simulations routine", submitted, **2021**. Available on ChemRxiv: 10.33774/chemrxiv-2021-h3381-v2.
24. Kevin Maik Jablonka, Fergus McIlwaine, Susana Garcia, Berend Smit, Brian Yoo, "A reproducibility study of 'Augmenting Genetic Algorithms with Deep Neural Networks for Exploring the Chemical Space'", arXiv 2102.00700, **2021**.
25. Kevin Maik Jablonka, Berend Smit, "Making MOFs designable—predicting the self-assembly of metal-organic frameworks", in preparation.
26. Kevin Maik Jablonka, Nancy Domingues, Berend Smit, "Data-driven discovery of metal-organic frame- works for benzene capture", in preparation.

27. María Victoria Gil, Kevin Maik Jablonka, Susana Garcia, Covadonga Pevida, Berend Smit, "A data-driven approach for finding optimal biomass gasification pathways", submitted.

Patents and books

28. Kevin Maik Jablonka: "Grundlagen der Thermodynamik für Studierende der Chemie", Springer Spektrum (Wiesbaden), 2017. ISBN: 978-3-658-17021-9.
29. Kevin Jablonka: "Verfahren zur Herstellung eines polymeren Kohlenstoffnitrid-Katalysators" (Process for the production of a polymeric carbon nitride catalyst). 2014. German patent (DE102014000888A1) granted, PCT application filed (WO2015110117A2).
30. Brian Yoo, Kevin Jablonka: "High throughput screening". 2021. (Appplication WO2022090579A1).