Thèse n° 9032

Structure-Property Relationships in Complex Materials by Combining Supervised and Unsupervised Machine Learning

Présentée le 12 juillet 2021

Faculté des sciences et techniques de l'ingénieur
Laboratoire de science computationnelle et modélisation
Programme doctoral en science et génie des matériaux

pour l'obtention du grade de Docteur ès Sciences

par

Benjamin Aaron HELFRECHT

Acceptée sur proposition du jury

Prof. V. Michaud, présidente du jury
Prof. M. Ceriotti, directeur de thèse
Prof. V. Van Speybroeck, rapporteuse
Prof. F.-X. Coudert, rapporteur
Prof. M. Dal Peraro, rapporteur

École
polytechnique
fédérale
de Lausanne

2021

Acknowledgements

This thesis would not have been possible without the immense help and guidance from my advisor, Prof. Michele Ceriotti. He always made himself available to discuss matters of science, research, and academia, for which I am eternally grateful.

I would also like to thank my co-authors, colleagues, and friends both within and outside of the the Laboratory of Computational Science and Modelling (COSMO) for their help and support over the last four years. In particular, I would like to thank Prof. Rocio Semino, Dr. Giovanni Pireddu, and Prof. Scott Auerbach for their constructive feedback, scientific input, and encouraging words over the course of our multi-year collaboration. I would additionally like to thank Alexander Goscinski and Dr. Andrea Anelli for lending me their individual expertise on the construction of the SOAP representation (Section 2.1.2) and the implementation of the generalized convex hull (Section 5.4), respectively, both of which were instrumental in the methodology applied in the capstone work of this thesis (Section 7.3). I would also like to extend a special thanks to Dr. Rose Cersonsky, who has been an incredible colleague, friend, and mentor; and to Anne Roy, whose hard work and dedication keeps the COSMO laboratory running like a well-oiled machine and makes it an incredibly welcoming place to work.

Finally, I would like to thank my parents Paul and Paula Helfrecht for their unwavering patience and support over the past four years. It's not easy being separated from family by 7,000 km and an ocean, and this thesis would not have been possible without their love and encouragement.

Abstract

Over the past decade, machine learning techniques have seen widespread adoption in the chemistry and materials science community, and for good reason: (continuing) advances in high performance computing have led to an explosion of public databases containing experimental, theoretical, and hypothetical materials and molecules alongside their observed (or predicted) properties. The result is a veritable sea of information that can be explored in search of new materials or to better understand those that already exist.

But just because we have vast quantities of data at our disposal does not mean that all of it is useful for every application. Part of the problem with "big data" is that it is so *big*. Sifting through a sea of real and hypothetical structures and properties is very much like finding a handful of needles among several dozen haystacks; brute force approaches quickly become nonviable. However, if we first consider a broad perspective of the available information, we can better understand the structure of the data space and subsequently determine where we should focus our efforts in order to find as many of the highest quality needles as possible. For instance, do we have reason to believe the needles will be distributed evenly among the different haystacks, and if not, which haystacks shall we search first in order to be the most efficient? Where in each haystack might we have the best chance of finding, not only the most, but also the highest quality needles for our target use case? Broadly speaking, posing and answering these kinds of questions is the realm of machine learning in the context of materials science and chemistry, and is the focus of this thesis.

In particular, the work presented in this thesis combines the two main paradigms of machine learning, namely *supervised learning*, where we attempt to predict certain characteristics of particular materials and molecules based on our knowledge about others, and *unsupervised learning*, in which we examine how materials and molecules are arranged in the data space to understand how they are related to one another, to examine structure–property relationships in databases of materials. The combination of these two approaches maps well onto the finding-needles-in-haystacks problem: through unsupervised learning, we are able to understand the layout of the haystacks; with supervised learning, we are able to narrow our search for useful needles and predict their ultimate quality. While either supervised learning or unsupervised learning alone can be a powerful tool for assessing materials and their properties, the focus here is to demonstrate the utility of *combining* both supervised and unsupervised learning to gain actionable insight about complex materials, whether through a unified approach or in sequential workflows. To this end, the application of combined supervised–unsupervised learning schemes will be presented for two examples, each focusing

on a different class of materials.

The first is an analysis of hydrogen bonding and backbone dihedral motifs in protein crystal structures from the Protein Data Bank, and demonstrates that data-driven definitions of structural motifs obtained through unsupervised learning can be more detailed and precise than conventional heuristics and can also be validated through supervised learning. We found that the motifs identified using a Gaussian mixture model largely agreed with more "traditional" definitions, but proved to be more precise for edge cases. Furthermore, we found that outside the more well-defined secondary structure motifs such as helices and sheets, several conventional secondary structure definitions did not coincide with the observed data-driven structural motifs, suggesting that the heuristic definitions corresponding to less-ordered secondary structure motifs do not strongly reflect the distribution of structural patterns in protein crystals in the Protein Data Bank; at the same time, there also exist clear, though as-yet unnamed motifs in the configuration space of proteins.

The second example centers around the exploration of structure–property relationships in all-silica zeolites, ultimately aiming to address the challenge of finding new zeolite frameworks that might be experimentally synthesizable. We begin by constructing a map of atom-centered environments in a database of hypothetical zeolite frameworks based on principal component analysis, where we validate our choice of "cardinal directions" by demonstrating that they correlate with the predicted atomic contributions to the molar volume and energy of the frameworks while emphasizing the diversity of the structural space. We extend this exploration of the structural space to a supervised classification exercise to distinguish hypothetical zeolite frameworks from those that have been experimentally synthesized, where frameworks that share several structural characteristics with synthesized frameworks are likely to be misclassified, and therefore may serve as promising synthesis candidates. To further filter the synthesis candidates based on their thermodynamic stability, we apply a convex hull construction based on a measure of classification prediction strength and the lattice energies of the zeolite frameworks. Through this combined supervised–unsupervised learning workflow we are able to propose a collection of hypothetical zeolites as likely candidates for experimental synthesis.

These two examples show that by combining supervised and unsupervised learning, it is possible to gain deeper insight into the structure–property relationships in a wide array of materials than through either set of methods in isolation, especially when using models and feature representations that allow for direct inspection of the structural characteristics that contribute most to the model outcomes. As the use of machine learning techniques in the materials science and chemistry community continues to grow, workflows and unified models that combine both supervised and unsupervised learning stand to become even more powerful tools for understanding structure–property relationships in materials and molecules.

Keywords: machine learning, supervised learning, unsupervised learning, structure–property relationships, hydrogen bonds, proteins, zeolites

Contents

List of Figures

List of Tables

1 Introduction

Throughout much of human history, innovations in materials have been closely tied to greater technological revolutions, from the birth of metallurgy to the dawn of the information age. In 2010, Christopher L. Magee made an attempt to quantify the contributions that materials innovation makes in the development of associated technologies, and concluded that between 20–80% of technological progress can be attributed to advances in materials, depending on the field [1]. However, the time between materials discovery and large-scale industrial adoption is on the order of several years to several decades [2, 3]. Given that there is a practically infinite number of possible materials and molecules, it is impossible to manually sift though all possible combinations of elements, bonding arrangements, crystal structures, and conformations to pick out those that might be good candidates for a particular application. Accelerating the technology transfer related to new materials, then, requires knowledge of structure–property relationships within the design space and *in silico* evaluations of materials that are capable of facilitating the development of synthesis and processing routes. Over the past decade, machine learning techniques have shown promise as tools for addressing these requirements, and have increasingly found use in materials discovery efforts.

As machine learning techniques have gained popularity in chemistry and materials science, they have been used to learn and predict energies and other ground-state quantities [4–14], crystal growth [15], mechanical properties [16], chemical shifts [17, 18], and dipole moments and polarizibilities [19–21], in addition to the electron density [22–25], density of states [25–27], and molecular wavefunctions [28–31]. Additionally, machine learning has been used to accelerate computational chemistry calculations through the generation of interatomic potentials [32–36], optimal basis functions [37], and by avoiding redundant computation of energies and forces of similar configurations in ab initio molecular dynamics simulations [38]. Finally, high-throughput techniques have been used to extract synthesis recipes from the literature [39–46] and to search for and evaluate materials for a particular application based on their structure and properties [47–49]. Concurrent with the growth of machine learning and statistical techniques in materials science research, a number of databases of hypothetical or real materials have come online, including those for small molecules [50–52], molecular crystals [53], framework materials [54–59], amino acid conformers [60, 61], and randomized structures including carbon polymorphs [62, 63], in addition to the continued growth of established databases such as the Protein Data Bank [64] and the Crystallography

Open Database [65–71].

To date, many of the applications of machine learning to materials science and chemistry could be classified as *supervised learning*, in which the goal is to predict a quantity $\mathbf{y}$ (e.g., a property or a categorical label) based on a representation of the known structure or properties $\mathbf{x}$. Each sample used to train a supervised learning model is thus associated with some *known* target data that we wish to reproduce for the training samples and accurately predict for new, unseen samples [72–74]. Perhaps less common are applications of *unsupervised learning*, where the goal is to analyze the structure of the data space in order to find statistically relevant motifs [72–74] (normally regarding the structure of different materials). Given the different paradigms offered by supervised and unsupervised learning, it can be instructive to apply both approaches concurrently to materials informatics problems: workflows that combine both learning paradigms allow us to identify (structural) motifs and subsequently examine their correlations with (predicted) materials properties in a unified manner.

This thesis aims to examine, both qualitatively and quantitatively, structure–property relationships in materials using combined supervised–unsupervised workflows and algorithms that can be leveraged to yield additional insight that would not be otherwise available. It is important to note that the hybrid supervised–unsupervised learning discussed in this thesis is distinct from what is often referred to as "semi-supervised learning", where one typically has goals similar to that of supervised learning (predicting properties or labels), but only some of the training data have known auxiliary labels or properties; the rest are missing [75]. By contrast, the hybrid supervised–unsupervised schemes that are the focus of this thesis combine supervised learning on fully labeled data with unsupervised learning that does not rely on any auxiliary information.

Chapter 2 discusses some of the practicalities of applying machine learning techniques to materials science and chemistry in particular, including specific considerations for feature representations and kernel methods. Chapters 3 and 4 discuss the specific unsupervised and supervised algorithms integral to this work, focusing on the principal component analysis and ridge regression families of methods. Chapter 5 presents methods for combining both supervised and unsupervised machine learning to extract structure–property relationships, detailing the concepts behind principal covariates regression and some of its nonlinear extensions. Using the techniques described in Chapters 2–5, Chapters 6 and 7 then provide examples of how supervised and unsupervised machine learning can be applied to chemistry and materials science, through examples of hydrogen bonding in proteins and searching for synthesizable zeolites. Specifically, Chapter 6 demonstrates the utility of unsupervised learning for identifying structural motifs in materials that can be validated through supervised learning, and Chapter 7 provides a thorough example of how supervised and unsupervised learning can be integrated into hybrid workflows that reveal structure–property relationships in a large database of structurally diverse materials.

2 Machine Learning for Materials and Molecules

When applying machine learning techniques to a particular collection of data, there are five basic steps that are generally employed, which are: (1) building a feature representation, (2) pre-processing the input data, (3) tuning the machine learning model, (4) training the model, and (5) evaluating the predictions made by the model. During the training process, the main objective is to minimize a loss function on a set of *training* samples; the particular form that this loss function takes is the main aspect that distinguishes different machine learning techniques from one another, and is discussed in more detail in Chapters 3–5, but is mentioned here as it will be revisited in Section 2.2. To start, this chapter discusses the practical aspects of this machine learning "recipe" that are particular to materials science and chemistry, namely the specific requirements of the feature representation and the modifications to the recipe that are required based on the nature of the structure–property relationships under investigation, i.e., whether they correspond to individual atomic environments or to whole structures. A brief introduction to kernel methods is also presented, as there are additional considerations that must be made when building kernels for materials data. More general aspects related to pre-processing and model tuning are discussed in Appendix A.

2.1 Feature Representations

The first step in applying machine learning methods to a collection of materials or molecules data is to compute a numerical representation in the form of a vector $\mathbf{x}$, often called the *feature representation*, for each structure (or atomic environment that the structure comprises) that can be understood and manipulated by the machine learning model. In principle, we have complete freedom to choose the form that $\mathbf{x}$ takes; however, not all choices are created equal. For instance, constructing $\mathbf{x}$ from a simple concatenation of the atomic coordinates is less than ideal, as the raw coordinates do not encode the symmetries and invariances that govern many structure–property relationships. The quality of the data-driven insights that we can derive from a machine learning model depends in large part on how well the feature representation is able to capture the similarities and differences between structures (or environments) in line with physical principles. In 2015, Ghiringhelli et al. [76] proposed a set of guidelines for materials descriptors that can be summarized as the following: (1) the feature representation must uniquely describe a structure (or environment), (2) the difference between two representations must be commensurate with the difference between the entities

that they describe, and (3) the representation is as low-dimensional as possible and the computation of the descriptor must not serve as the bottleneck in a machine learning pipeline. Consequently, much effort in the field of materials informatics has been devoted to the design and implementation of materials-specific feature representations.

2.1.1 The Feature Representation Zoo

Materials and molecules can be represented in machine learning models in a number of ways, including approximate interaction representations like the Coulomb Matrix [4], Bag of Bonds [6], and other related representations [5, 77, 78]; molecular graphs [11, 13]; Voronoi tesselations [79]; representations based on radial distribution functions [12, 27]; character strings such as SMILES [80, 81]; a vector of properties [18]; stoichiometry and crystal sites [7]; bonding motifs [10]; and vectorized descriptions of the atomic structure such as the Faber-Christensen-Huang-von Lilienfeld representation [9], Many-Body Tensor Representation [82], Behler-Parrinello symmetry functions [83], and the Smooth Overlap of Atomic Positions (SOAP) [84–89]. Each feature representation has its own advantages and disadvantages, ranging from its descriptive power, computational expense, and uniqueness, to its embedded symmetries and invariances. The latter is of particular importance, given that the structures and properties of molecules and materials exhibit certain translational, rotational, and permutation symmetries and invariances. For example, the energy of a molecule does not change if it is rotated or translated in space. While this may seem obvious, this sort of intuition is not automatically understood by machine learning models. To acquire predictions that take these symmetries and invariances into account, they must be baked into the machinery of the model itself, encoded into the feature representation, or learned through large quantities of data. In the example given above, the translational and rotational invariance of the energy could be learned by training on a large number of identical molecules that differ only in their relative orientation in space. Building models in this way can be effective, but it is rather inefficient and is often avoided in favor of using feature representations or purpose-built models that incorporate the relevant physics.

The feature representation must also be chosen to be compatible with the problem at hand. For example, a descriptor constructed by concatenating a number of properties contains no explicit structural information and therefore may not be particularly useful for exploring structure–property relationships. Similarly, descriptors that depend only on connectivity, such as SMILES- or graph-based representations, may not be as useful for investigating differences between different conformers of the same molecule, and descriptors that do not account for periodic boundary conditions, such as the Coulomb Matrix or Bag of Bonds, are not applicable to crystals.

The work in this thesis uses primarily the SOAP representation, as it is a versatile and generally applicable framework for describing both two- and three-body correlations in periodic and non-periodic structures, and has seen great success in predicting molecular properties [90], recognizing structural motifs [90, 91], and constructing machine-learning-based interatomic potentials [32–36, 92].

2.1.2 Smooth Overlap of Atomic Positions

The SOAP approach generates a feature representation for a structure from an atomic density constructed by describing each atom i with an individual Gaussian density $g(\mathbf{r}-\mathbf{r}_i)$ [84–89]. Following the notation of Refs. [86–89], the atomic density associated with an environment A_j centered on atom j within a structure A can be written as [86, 88]

$$\langle \mathbf{r} | A_j \rangle = \sum_{i \in A} f_c(r_{ij}) g(\mathbf{r}-\mathbf{r}_{ij}) \, |a_i\rangle \,, \tag{2.1}$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $r_{ij} = \|\mathbf{r}_{ij}\|$, f_c is a smooth cutoff function describing the spatial limits of the environment around atom j, and $|a_i\rangle$ is an attribute representing the species of each atom. The atomic environment corresponding to a single "species channel" can be written as a sum over only those atoms of a particular species a [86–88]

$$\langle a\mathbf{r} | A_j \rangle = \sum_{i \in a} f_c(\mathbf{r}_{ij}) g(\mathbf{r}-\mathbf{r}_{ij}), \tag{2.2}$$

where $i \in a$ indicates all atoms i in structure A having species a, so that the representation of the environment can additionally be represented as a sum of contributions from individual atomic species. An atomic environment $\langle \mathbf{r} | A_j \rangle$ constructed in this way can be made translationally invariant through Haar integration over translations [86–88], and a description of the whole structure can be represented as a sum over these atom-centered environments [88],

$$|A\rangle = \sum_j |a_j\rangle \otimes |A_j\rangle \,. \tag{2.3}$$

In practical situations, it is convenient to expand the density in a basis of radial functions $R_n(\mathbf{r})$ (for example, Gaussian-type orbitals or polynomials in the discrete variable representation [93]) and spherical harmonics $Y^l_m(\hat{\mathbf{r}})$, where $\hat{\mathbf{r}} = \mathbf{r}/\|\mathbf{r}\|$, to avoid convergence issues based on the discretization of the continuous, real-space representation. The density coefficients of such an expansion are [86–88]

$$\langle anlm | A_j \rangle = \int \mathrm{d}\mathbf{r} R_n(\mathbf{r}) Y^l_m(\hat{\mathbf{r}}) \langle a\mathbf{r} | A_j \rangle \,. \tag{2.4}$$

Similar to incorporating translation invariance, $(\nu+1)$-body rotationally invariant representations of the environment, denoted $\left| A_j^{(\nu)} \right\rangle$, can be constructed from Haar integration over rotations [86–88] and simplified into summations over the density coefficients arising from averaging a tensor product of ν environment descriptors over the $SO(3)$ rotation group [86–88]. A two-body rotationally invariant representation (referred to as the *radial spectrum*) can thus be written as [88]

$$\left\langle an \middle| A_j^{(1)} \right\rangle = \langle an00 | A_j \rangle \,, \tag{2.5}$$

and is analogous to the radial distribution function for each pair of atomic species composing each atom-centered environment. Similarly, a three-body rotationally invariant representation

(the SOAP *power spectrum*) for A_j can be written as [86–88],

$$\left\langle ana'n'l \middle| A_j^{(2)} \right\rangle = \frac{1}{\sqrt{2l+1}} \sum_m (-1)^m \left\langle anlm \middle| A_j \right\rangle \left\langle a'n'l(-m) \middle| A_j \right\rangle. \tag{2.6}$$

The SOAP power spectrum can be viewed as a "template" or "stencil" comprising two line segments of lengths r and r' sharing a vertex at the location of atom j and being separated by an angle θ [88]. The power spectrum is the result of integrating over all rotations and configurations of this template within the atomic density centered at the shared vertex of the two line segments.

In a similar fashion, it is possible to recover the real-space atomic density [88] from the radial spectrum,

$$\left\langle a\mathbf{r} \middle| A_j^{(1)} \right\rangle = \sum_n R_n(\mathbf{r}) \left\langle an \middle| A_j^{(1)} \right\rangle \tag{2.7}$$

and the power spectrum,

$$\left\langle a\mathbf{r}a'\mathbf{r}'\omega \middle| A_j^{(2)} \right\rangle = \sum_{n,n',l} R_n(\mathbf{r}) R_{n'}(\mathbf{r}') P_l(\omega) \left\langle ana'n'l \middle| A_j^{(2)} \right\rangle, \tag{2.8}$$

where $\omega = \hat{\mathbf{r}} \cdot \hat{\mathbf{r}}' = \cos\theta$. The transformation to a real-space representation can be employed in the context of a machine learning model, where the coefficients, or weights, of the trained model can be expanded into a real-space representation in the same way as the SOAP vectors, making it possible to transparently correlate structural features with the model weights.

The complexity of the SOAP representation makes it relatively expensive to compute; moreover, many of the SOAP spectrum features can be considered redundant with one another to some degree. It can thus be useful to construct a "contracted" SOAP representation based on an augmented set of radial basis functions that are constructed by computing the covariance of the density coefficients for each species and angular channel. The eigenvectors of this covariance are then used to project the radial basis functions [94], which can then be used to compute features through a spline-based fit [93], reducing the computational expense of the SOAP representation while retaining a high-quality radial basis. This procedure can be used to reduce certain artifacts and improve interpretability in the real-space expansion.

2.2 Kernel Methods

Even if we choose a feature representation that accurately, completely, and uniquely describes the input structures, it is still often the case that there exist complex, nonlinear relationships between samples or between the features and prediction targets that are not immediately apparent in the raw feature space. One way to transparently disentangle these relationships is to apply a nonlinear transformation to each sample of the input data $\mathbf{x}_i$ through a function ϕ that maps the input data to a high- (or potentially infinite-) dimensional space where we can then apply linear learning techniques. We denote the representation of each sample in the high-dimensional space as $\boldsymbol{\phi}_i = \phi(\mathbf{x}_i)$. Such a transformation can be advantageous, for example, in classification exercises, where data that is not linearly separable in the original space

may become linearly separable in the high-dimensional space [95]. However, working directly in the high-dimensional space can be impractical, or even impossible where the dimension of the space is infinite. We can make the problem of operating in the high-dimensional feature space more tractable by framing the exercise of finding feature–target relationships in terms of assessing the similarity between different samples [95]: when we make a prediction for a particular feature vector, we are effectively comparing it against the training samples and interpolating accordingly. One simple metric to quantify the similarity between two samples is the dot product [95], which further allows us to reformulate the mapping $\mathbf{x}_i \mapsto \boldsymbol{\phi} = \phi(\mathbf{x}_i)$ in terms of a *kernel function* k that describes the similarity between two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ in the high-dimensional space [95],

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j). \tag{2.9}$$

In this way, we can avoid dealing directly with ϕ, and instead construct a function k that operates on the original feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. The tradeoff is that not all functions can be expressed in the form of Eqn. 2.9. Fortunately, the kernel functions that do admit such an expression are given by Mercer's theorem [96]: if the Gramian matrix $\mathbf{K}$ of a kernel function, whose entries are $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, is positive definite, then the kernel can be expressed as a dot product [95]. Common kernels include the *polynomial kernel* [95]

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + c)^d, \tag{2.10}$$

and the *radial basis function kernel* [95]

$$k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}. \tag{2.11}$$

Another commonly used kernel is the *linear kernel*, which is a special case of the polynomial kernel with $c = 0$ and $d = 1$, so that $\phi(\mathbf{x}_i) = \mathbf{x}_i$. According to the Moore-Aronszajn Theorem [97], each kernel has associated with it a unique vector space known as the *reproducing kernel Hilbert space* (RKHS) $\mathscr{H}$, wherein [95]

$$\phi(\cdot)^T k(\mathbf{x}, \cdot) = \phi(\mathbf{x}), \qquad \forall \phi \in \mathscr{H} \tag{2.12}$$

and k spans the RKHS [95]. Furthermore, the Representer theorem [98–100], states that the minimum of a loss function in an RKHS corresponding to a given kernel can be expressed as a linear combination of kernel values evaluated at the training points [95]. Therefore, we need not operate directly in the high-dimensional RKHS or know the mapping $\phi(\mathbf{x})$ explicitly in order to perform nonlinear learning exercises; we need only know the kernel function. The use of this kernel construction will be applied to several of the algorithms presented in Chapters 3–5 and is used in several of the machine learning exercises presented in Chapters 6 and 7.

2.3 Representative Atomic Environments

While kernel methods facilitate straightforward and transparent nonlinear data analyses, they can be rather expensive to carry out on datasets with a large number of samples, as the number of entries in the kernel matrix (for the training data) grows with the square of the number of samples. To reduce memory requirements, it is possible to use the Nyström approximation [101] to build a low-rank approximation to the full kernel matrix by using a subset of the samples to serve as a set of representatives or a reference set. In the context of materials applications, the reference set comprises representations of individual atomic environments. Given that the kernel matrix is symmetric and positive semi-definite, it possesses an eigendecomposition $\mathbf{K} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$, where the eigenvalues, as the entries of the diagonal matrix $\boldsymbol{\Lambda}$, are real and non-negative. We can thus build a low-rank approximation $\hat{\mathbf{K}}$ to the full kernel matrix $\mathbf{K}$ through an eigendecomposition of a smaller kernel matrix based on a subset of M of the N total samples used to build the full kernel [101]. If we define $\hat{\mathbf{K}} = \hat{\mathbf{U}}\hat{\boldsymbol{\Lambda}}\hat{\mathbf{U}}^T$, then the quantities $\hat{\mathbf{U}}$ and $\hat{\boldsymbol{\Lambda}}$, which are approximations to the eigenvectors and eigenvalues of $\hat{\mathbf{K}}$, can be written in terms of the eigendecomposition of the kernel between the M representative points $\mathbf{K}_{MM}$ [101],

$$\hat{\boldsymbol{\Lambda}} = \frac{N}{M}\boldsymbol{\Lambda}_{MM} \tag{2.13}$$

$$\hat{\mathbf{U}} = \sqrt{\frac{M}{N}}\mathbf{K}_{NM}\mathbf{U}_{MM}\boldsymbol{\Lambda}_{MM}^{-1}, \tag{2.14}$$

where $\boldsymbol{\Lambda}_{MM}$ and $\mathbf{U}_{MM}$ are the eigenvalues and eigenvectors of $\mathbf{K}_{MM}$, and $\mathbf{K}_{NM}$ is the kernel matrix between the full set of N samples and the M representative samples. Therefore, we can write the Nyström approximation as [101]

$$\mathbf{K} \approx \hat{\mathbf{K}} = \mathbf{K}_{NM}\mathbf{K}_{MM}^{-1}\mathbf{K}_{NM}^T. \tag{2.15}$$

Through the Nyström approximation, we can also compute an approximation to the RKHS features, by noting that $\mathbf{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T$, so that $\hat{\mathbf{K}} = \boldsymbol{\Phi}_{NM}\boldsymbol{\Phi}_{NM}^T$, where

$$\boldsymbol{\Phi}_{NM} = \mathbf{K}_{NM}\mathbf{U}_{MM}\boldsymbol{\Lambda}_{MM}^{-1/2}, \tag{2.16}$$

and the rows of the matrix $\boldsymbol{\Phi}$ are the individual RKHS feature vectors $\boldsymbol{\phi}_i$.

The representative samples can be selected in a number of ways, e.g., randomly [101], probabilistically [102, 103], iteratively [104], or through k-means clustering [105]. Farthest point sampling (FPS) [106] is a particularly useful selection method in the context of the Nyström approximation, as it yields a diverse subset of samples. In FPS, samples are selected iteratively, with the goal of maximizing the distance between a new sample and those already selected. In particular, the point to be selected is that with the largest minimum distance to any of the samples that have already been selected. This can be achieved through the iteration

[106],

$$\mathbf{s}_{n+1} = \underset{\mathbf{x}_i \in X}{\operatorname{argmax}} \left[\min_{\mathbf{s}_j \in S_n} d(\mathbf{x}_i, \mathbf{s}_j) \right]$$
$$S_{n+1} = S_n \cup \mathbf{s}_{n+1}$$

where $d(\mathbf{x}_i, \mathbf{s}_j)$ is the distance between the samples $\mathbf{x}_i$ and $\mathbf{s}_j$, S_n is the set of previously selected samples, and S_{n+1} the set of selected samples with the newest point $\mathbf{s}_{n+1}$ added. The first point is typically chosen at random. The FPS procedure can also be used to select a diverse set of features and can be a relatively simple way to reduce the dimensionality of a dataset.

2.4 Structures and Environments

One final aspect in the application of machine learning to materials and molecules that warrants discussion is that of property and model additivity. Based on the particular structure–property relationships we wish to explore, we can take a local, atom-centered approach, or a global, structure-based approach. For those global properties that can be expressed as a sum of contributions from individual atomic environments, i.e., $\mathbf{y}_A = \sum_{A_j \in A} \mathbf{y}_{A_j}$, we can make equivalent predictions in linear models for the global property by summing over the model predictions for the individual environments composing the global structure, or by training and evaluating the model on global features that themselves are a sum over the individual environment-based features, namely [107],

$$\mathbf{x}_A = \sum_{A_j \in A} \mathbf{x}_{A_j}. \tag{2.17}$$

An analogous procedure can be adopted for kernel-based models, where the kernel between two structures can be expressed as the sum of the kernel values between the individual environments composing the structures [14, 85, 87],

$$\mathbf{K}_{AB} = \sum_{i \in A_i} \sum_{j \in B} k(\mathbf{x}_{A_i}, \mathbf{x}_{B_j}). \tag{2.18}$$

One can similarly define a kernel between a structure and a set of environments by performing the sum only over the environments in the structure of interest, i.e. [108],

$$\mathbf{K}_{Aj} = \sum_{i \in A_i} k(\mathbf{x}_{A_i}, \mathbf{x}_{\bullet_j}), \tag{2.19}$$

where the environment-based features $\mathbf{x}_{\bullet_j}$ can come from multiple structures.

3 Unsupervised Learning [1]

Unsupervised learning generally involves learning the structure of the feature space in order to make statements about how the underlying samples are related to one another. Unsupervised learning comes in many different forms, with two of the most common categories being *clustering*, or *motif recognition*, and *dimensionality reduction*. The following sections provide an overview of the learning techniques in these two categories that are relevant to the work presented in Chapters 6 and 7.

3.1 Motif Identification

Broadly speaking, unsupervised clustering techniques can be considered a means of recognizing patterns, or motifs, within a feature space, where the goal is typically to subdivide the feature space into distinct regions with each region corresponding to a separate motif. While this is conceptually similar to supervised classification exercises, discussed further in Chapter 4, the main difference is that in clustering techniques the subdivision of the feature space is made without the use of any auxiliary labels associated with the data.

Common clustering techniques include k-means [72–74], hierarchical clustering [72, 74], self-organizing maps (also called Kohonen maps) [74, 109], and mixture models [72–74]. The following subsection describes the Probabilistic Analysis of Molecular Motifs (PAMM) algorithm [110, 111], which is used in Chapter 6 to identify hydrogen bonding and backbone dihedral motifs in protein structures.

3.1.1 Probabalistic Analysis of Molecular Motifs

The PAMM algorithm is an automated and flexible clustering technique based on a Gaussian mixture model that was originally developed to identify motifs in materials, but can easily be applied to any domain.

Given a set of samples $\mathbf{X}$, the PAMM algorithm builds an approximation of the probability density $P(\mathbf{x})$ in the feature space based on a kernel density estimation (KDE) [72–74] on a

[1] Sections 3.2.1–3.2.4 of this chapter are adapted with modifications under the Creative Commons Attribution 4.0 (CC BY 4.0) license from Helfrecht, B. A., Cersonsky, R. K., Fraux, G. & Ceriotti, M. Structure-Property Maps with Kernel Principal Covariates Regression. *Machine Learning: Science and Technology* **1,** 045021. doi:10.1088/2632-2153/aba9ef (2020); all authors contributed to the writing of the manuscript from which the present text has been adapted.

subset of the samples $\mathbf{X}'$ selected through FPS (see Section 2.3 for a brief summary of FPS). In principle, any kernel can be used for the density estimation, and one common choice is the Gaussian kernel [2] [110, 111]

$$K(\mathbf{x};\mathbf{H}) = \frac{1}{\sqrt{(2\pi)^D \det(\mathbf{H})}} \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{H}^{-1}\mathbf{x}\right), \tag{3.1}$$

where $\mathbf{H}$ is a matrix of kernel bandwidths and D the dimensionality of $\mathbf{x}$, so that the approximate probability density at a point $\mathbf{x}'_i$ is [110, 111]

$$P(\mathbf{x}'_i) = \frac{\sum_{j=1}^{N} w_j K(\mathbf{x}_j - \mathbf{x}'_i;\mathbf{H}_j)}{\sum_{j=1}^{N} w_j}, \tag{3.2}$$

where the individual kernels can be assigned weights w_j. The bandwidth matrix $\mathbf{H}$ can be chosen in a number of ways. The simplest approach is to construct $\mathbf{H}$ as a diagonal matrix where the entries H_{jj} are chosen manually or based on a Voronoi decomposition of the sample points $\mathbf{X}'$, in which case the bandwidth H_{jj} associated with the point $\mathbf{x}'_j$ within a given Voronoi cell is set to the distance between the center of the parent Voronoi cell and the nearest neighboring Voronoi center [110]. The bandwidth matrix can also be based on a local estimate of the covariance around a given point through a heuristic such as Silverman's Rule [111],

$$\mathbf{H}_i = \left(\frac{4}{N_i(D_i+2)}\right)^{2/D_i+4} \mathbf{C}_i, \tag{3.3}$$

where N_i, D_i, and $\mathbf{C}_i$ are estimates of the local population, dimensionality, and covariance. The local population N_i is defined as a sum of local weights [111]

$$N_i = \sum_j u_{ij}, \tag{3.4}$$

where

$$u_{ij} = \frac{N w_j}{\sum_j w_j} \exp\left(-\frac{(\mathbf{x}_j - \mathbf{x}_i)^T(\mathbf{x}_j - \mathbf{x}_i)}{2\sigma_i^2}\right) \tag{3.5}$$

and the σ_i are tunable localization parameters. The local covariance $\mathbf{C}_i$ is computed as a (biased) weighted covariance from the weights u_{ij} and the oracle approximating shrinkage estimator [111, 112]. The local dimensionality D_i is then estimated from the eigenvalue spectrum of the local covariance [111]

$$D_i = \exp\left(-\sum_{k=1}^{D} \eta_k \log(\eta_k)\right), \tag{3.6}$$

[2] The use of the term *kernel* here is distinct from usage related to *kernel methods* discussed elsewhere in this thesis. In the context of the KDE, the kernel refers to the function used to transform the discrete samples into a continuous density.

where $\eta_k = \lambda_k / \left(\sum_{k=1}^{D} |\lambda_k|\right)$ and λ_k are the eigenvalues of the local covariance before applying the shrinkage estimator. To reduce computational expense, the bandwidth for a point $\mathbf{x}_j$ falling within the Voronoi cell of a sample point $\mathbf{x}'_i$ can be set to that computed for $\mathbf{x}'_i$.

Once the KDE on the sample points has been constructed, the sample points can be divided into k clusters, for example, using the quick-shift algorithm [110, 111, 113]. After the cluster assignments are made, the probability density of the samples in the feature space $P(\mathbf{x})$ can be approximated through a Gaussian mixture model, where the approximate density $\hat{P}(\mathbf{x})$ is constructed as a weighted sum of (normalized) Gaussians centered on the cluster modes $\boldsymbol{\mu}_k$ [110, 111],

$$\hat{P}(\mathbf{x}) = \sum_{k=1}^{n} p_k G(\mathbf{x}; \boldsymbol{\mu}_k, \mathbf{C}_k), \tag{3.7}$$

where the covariance of cluster $\mathbf{C}_k$ is determined based on the value of the KDE evaluated at the sample points $\mathbf{x}'_i$ belonging to cluster k. The PAMM algorithm can be extended to periodic feature spaces by replacing the Gaussian functions G in Eqn. 3.7 with a product of one-dimensional von Mises distributions [111]. Given a mixture of Gaussians, the probability that a given sample $\mathbf{x}$ belongs to a cluster k is [110, 111],

$$\hat{P}_k(\mathbf{x}) = \frac{p_k G(\mathbf{x}; \boldsymbol{\mu}_k, \mathbf{C}_k)}{\hat{P}(\mathbf{x}) + \zeta}, \tag{3.8}$$

which can be used to construct a "fingerprint" for each cluster. The parameter ζ acts as a cutoff density for outlier configurations.

3.2 Dimensionality Reduction

Another common use for unsupervised learning is dimensionality reduction, where the aim is to construct a new, condensed feature space (often referred to in the following as the *latent space*) from the original set of features while minimizing information loss. There are a wide variety of dimensionality reduction techniques, including locally linear embedding [114], Isomap [115], stochastic neighbor embedding [116] and its successor t-distributed stochastic neighbor embedding [117], and density-based spatial clustering of applications with noise (DBSCAN) [118] and its extension HDBSCAN [119, 120]. The machine learning applications presented in Chapters 6 and 7 make use of the principal component analysis (PCA) family of methods in addition to sketch-map [121–123], and these methods are described in the following subsections.

3.2.1 Principal Component Analysis

In principal component analysis (PCA) [124, 125], the aim is to reduce the dimensionality of a centered feature matrix $\mathbf{X}$ (which contains the individual samples $\mathbf{x}_i$ as rows) by constructing a low-dimensional, orthogonal projection $\mathbf{T} = \mathbf{X}\mathbf{P}_{XT}$ that results in minimal information loss when $\mathbf{X}$ is projected into the low-dimensional space and back. This is equivalent to minimizing

the loss

$$\ell = \|\mathbf{X} - \mathbf{X}\mathbf{P}_{XT}\mathbf{P}_{TX}\|^2 \tag{3.9}$$

with respect to $\mathbf{P}_{XT}$, with the additional constraint that $\mathbf{P}_{XT}$ is orthogonal. This constraint implies $\mathbf{P}_{TX} = \mathbf{P}_{XT}^T$, so that the loss can be rewritten as

$$\ell = \mathrm{Tr}\left(\mathbf{X}\left(\mathbf{I} - \mathbf{P}_{XT}\mathbf{P}_{XT}^T\right)\mathbf{X}^T\right), \tag{3.10}$$

or equivalently cast as a maximization of the similarity

$$\rho = \mathrm{Tr}\left(\mathbf{P}_{XT}^T\mathbf{X}^T\mathbf{X}\mathbf{P}_{XT}\right). \tag{3.11}$$

Because $\mathbf{P}_{XT}$ and $\mathbf{P}_{TX}$ are orthogonal, the similarity is maximised when $\mathbf{P}_{XT}$ is the matrix of eigenvectors of the covariance $\mathbf{C} = \mathbf{X}^T\mathbf{X}$ that are associated with the n_{latent} largest eigenvalues. Writing the eigendecomposition as $\mathbf{C} = \mathbf{U}_\mathbf{C}\mathbf{\Lambda}_\mathbf{C}\mathbf{U}_\mathbf{C}^T$, where $\mathbf{\Lambda}_\mathbf{C}$ is the diagonal matrix of the eigenvalues and $\mathbf{U}_\mathbf{C}$ is the matrix containing the corresponding eigenvectors as its columns, the orthogonal projection $\mathbf{T}$ is

$$\mathbf{T} = \mathbf{X}\hat{\mathbf{U}}_\mathbf{C}, \tag{3.12}$$

where $\hat{\mathbf{U}}_\mathbf{C}$ is the submatrix of $\mathbf{U}_\mathbf{C}$ containing the first n_{latent} columns of $\mathbf{U}_\mathbf{C}$, that is, the eigenvectors corresponding to the largest eigenvalues of $\mathbf{C}$.

3.2.2 Kernel Principal Component Analysis

One can also perform principal component analysis in the RKHS by substituting for $\mathbf{X}$ the matrix of (centered) transformed RKHS features $\mathbf{\Phi}$. If we wish to avoid explicit computation of $\mathbf{\Phi}$, we can take advantage of the fact that the covariance $\mathbf{C} = \mathbf{\Phi}^T\mathbf{\Phi}$ and the kernel $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T$ have the same (nonzero) eigenvalues and that the eigenvectors $\mathbf{U}_\mathbf{K}$ can be written in terms of those of $\mathbf{U}_\mathbf{C}$ through the expression $\mathbf{U}_\mathbf{K} = \mathbf{\Phi}\mathbf{U}_\mathbf{C}\mathbf{\Lambda}_\mathbf{C}^{-1/2}$, so that the projections $\mathbf{T}$ may be written in terms of the decomposition of the kernel matrix,

$$\mathbf{T} = \hat{\mathbf{U}}_\mathbf{K}\hat{\mathbf{\Lambda}}_\mathbf{K}^{1/2} = \mathbf{K}\hat{\mathbf{U}}_\mathbf{K}\hat{\mathbf{\Lambda}}_\mathbf{K}^{-1/2}. \tag{3.13}$$

This approach is known as kernel principal component analysis (KPCA) [126]. From inspection of Eqn. 3.13, we can define the matrix $\mathbf{P}_{KT} = \hat{\mathbf{U}}_\mathbf{K}\hat{\mathbf{\Lambda}}_\mathbf{K}^{-1/2}$. We can additionally approximate the kernel from the latent space via the linear regression solution $\mathbf{P}_{TK} = \hat{\mathbf{\Lambda}}_\mathbf{K}^{1/2}\hat{\mathbf{U}}_\mathbf{K}^T$, and notice that in the latent space the kernel $\mathbf{K}$ is approximated by $\mathbf{T}\mathbf{T}^T$. If $\mathbf{K}$ is the linear kernel, KPCA reduces to PCA.

3.2.3 Low-Rank Kernel Principal Component Analysis

We can also perform KPCA using a low-rank approximation to the kernel matrix $\mathbf{K}$ through the Nyström approximation, where the covariance is constructed from the centered approximate

RKHS features $\mathbf{\Phi}_{NM}$,

$$\mathbf{C} = \mathbf{\Phi}_{NM}^T \mathbf{\Phi}_{NM} = \mathbf{U}_\mathbf{C} \mathbf{\Lambda}_\mathbf{C} \mathbf{U}_\mathbf{C}^T. \tag{3.14}$$

The projections are then computed just as in standard PCA,

$$\mathbf{T} = \mathbf{\Phi}_{NM} \hat{\mathbf{U}}_\mathbf{C} = \mathbf{K}_{NM} \mathbf{U}_{\mathbf{K}_{MM}} \mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2} \hat{\mathbf{U}}_\mathbf{C}, \tag{3.15}$$

so that $\mathbf{P}_{KT} = \mathbf{U}_{\mathbf{K}_{MM}} \mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2} \hat{\mathbf{U}}_\mathbf{C}$.

3.2.4 Multidimensional Scaling

A different approach to dimensionality reduction is that of multidimensional scaling (MDS) [127]. In MDS, the latent space is chosen to preserve the pairwise distances of the original feature space, which corresponds to the loss

$$\ell = \sum_{i<j} \left(d(\mathbf{x}_i, \mathbf{x}_j) - d(\mathbf{t}_i, \mathbf{t}_j) \right)^2, \tag{3.16}$$

where $\mathbf{x}_i$ and $\mathbf{t}_i$ refer to the full and projected feature vector of the ith sample and d is a general distance function. In *classical MDS*, d is taken to be the Euclidean distance and a modified loss is used,

$$\ell = \|\mathbf{K} - \mathbf{T}\mathbf{T}^T\|^2, \tag{3.17}$$

the minimization of which is equivalent to KPCA with a linear kernel (and thus standard PCA).

3.2.5 Sketch-Map

Sketch-map [121, 123] is an embedding method for dimensionality reduction that is based on the same general idea as MDS and has been used to construct enhanced sampling methods for molecular dynamics [122]. Rather than directly minimizing the differences between the samplewise distances in the high- and low-dimensional spaces as in MDS, sketch-map aims to minimize the differences between samplewise distances r in the high- and low-dimensional spaces that are transformed through a sigmoid function s [121, 123],

$$s(r; \sigma, a, b) = 1 - (1 + (r/\sigma)^a (2^{a/b} - 1))^{-b/a}, \tag{3.18}$$

where a, b, and σ are adjustable parameters. The sketch-map projection is thus determined by minimizing the loss [121–123]

$$\ell = \frac{\sum_{i<j} \left(s(x_{ij}; \sigma, a_X, b_X) - s(t_{ij}; \sigma, a_T, b_T) \right)^2}{\sum_{i<j} w_i w_j}, \tag{3.19}$$

where w_i and w_j are individual sample weights, $x_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$ is the distance between points $\mathbf{x}_i$ and $\mathbf{x}_j$ in the high-dimensional space, and $t_{ij} = d(\mathbf{t}_i, \mathbf{t}_j)$ is the distance between points $\mathbf{t}_i$ and $\mathbf{t}_j$ in the low-dimensional space. The choice of the parameter σ is largely based on the

length scale of the features that one wishes to emphasize in the embedding construction, and should generally be chosen so that the inflection point of the sigmoid is placed at a slightly smaller distance than where the first major peak occurs in a histogram of intersample distances in the high-dimensional space [123]. Separate a and b parameters are typically used for the high-dimensional space (subscript X) and the low-dimensional space (subscript T); the choice of these parameters is less critical, and Ref. [123] provides some heuristics for their selection.

Once the low-dimensional projections $\mathbf{t}_i$ have been determined through the minimization, a new point $\mathbf{x}'_k$ can be projected into the low-dimensional space by minimizing [121, 123]

$$\ell' = \frac{\sum_i w_i \left(s(x'_{ik};\sigma, a_X, b_X) - s(t'_{ik};\sigma, a_T, b_T)\right)^2}{\sum_i w_i}. \tag{3.20}$$

These samplewise minimization procedures can be computationally expensive for large numbers of samples; therefore, it can be advantageous to only compute the distances and projections for a small number of representative points. Farthest point sampling is again an effective method for selecting the representative points [123].

4 Supervised Learning [1]

The goal of supervised learning is to assign a target quantity or label to an unlabeled sample based on knowledge about a set of known (*feature vector*, *target*) pairs. When the targets can take on a continuous range of (usually numeric) values, regression techniques are typically applied. When the targets are categorical, classification algorithms can be used to assign discrete labels to the individual samples. Several classification algorithms, such as logistic regression [72–74] and support vector machines [128] assign class labels based on the thresholding of a continuous output. Other methods, such as linear discriminant analysis [72–74] can be used for dimensionality reduction or as a simple classifier. Still other methods, including decision trees [72–74] and k-nearest neighbors [72–74], can be applied to either regression or classification. In Chapters 6 and 7 we make use of various regression techniques as well as support vector machines. The relevant methods are described in the following sections.

4.1 Regression

Regression techniques are arguably the most familiar and straightforward supervised machine learning methods. Given a set of samples $\mathbf{X}$, we aim to build a model that is able to accurately reproduce the the corresponding targets $\mathbf{Y}$ by minimizing the difference between the predicted targets $\hat{\mathbf{Y}}$ and the actual targets $\mathbf{Y}$. At the same time, we would like the model to be capable of making accurate predictions for new inputs $\mathbf{X}'$ that the model has never seen before. If we stipulate that the model predict the known targets $\mathbf{Y}$ as accurately as possible, it is likely that the model will not be as accurate in making predictions for unseen samples. To illustrate this, suppose that we have a set of prediction targets $\mathbf{y}_i$ perturbed by some additive random noise ϵ_i normally distributed with mean zero, i.e., $\tilde{\mathbf{y}}_i = \mathbf{y}_i + \epsilon_i$. If we train a model to achieve perfect accuracy on the noisy targets $\tilde{\mathbf{y}}_i$, the predictions it makes for out-of-sample data will be strongly influenced by the particular draw of the ϵ_i. Consequently, if we train an ensemble of models by repeatedly re-rolling the ϵ_i and enforcing perfect predictions on the resulting noisy targets $\tilde{\mathbf{y}}_i$, we will observe a large variance in the out-of-sample predictions among

[1] Sections 4.1.1–4.1.3 of this chapter are adapted with modifications under the Creative Commons Attribution 4.0 (CC BY 4.0) license from Helfrecht, B. A., Cersonsky, R. K., Fraux, G. & Ceriotti, M. Structure-Property Maps with Kernel Principal Covariates Regression. *Machine Learning: Science and Technology* **1,** 045021. doi:10.1088/2632-2153/aba9ef (2020); all authors contributed to the writing of the manuscript from which the present text has been adapted.

the different models [72]. To address this issue, we can add to the regression loss function a penalty on the norm of the regression weights, thus encouraging the weights to be small and reducing the variance by introducing a bias. We must take care, however, that we do not penalize the weights too harshly, as a large bias can also reduce the accuracy of our predictions [72]. As a result, the magnitude of the penalty is typically left as a tunable parameter subject to optimization during the model tuning process (see Appendix A). A penalty based on the $L2$ norm of the weights is known as *ridge* or *Tikhonov* [129] regularization, and is perhaps the most common form of regularization. Penalizing instead based on the $L1$ norm of the regression weights forms the least absolute shrinkage and selection operator (LASSO) [130]. The LASSO tends to assign nonzero weight to only a subset of the features, and can be used to mark certain features as important for predicting the targets. Elastic net regularization [131] is a combination of $L1$ and $L2$ regularization. In the following, we focus solely on $L2$ regularization.

4.1.1 Ridge Regression

In linear regression, our goal is to determine a set of weights $\mathbf{P}_{XY}$ such that the difference between the known targets $\mathbf{Y}$ and the corresponding predicted targets $\hat{\mathbf{Y}} = \mathbf{X}\mathbf{P}_{XY}$ is minimized. This corresponds to minimizing the loss

$$\ell = \|\mathbf{Y} - \mathbf{X}\mathbf{P}_{XY}\|^2. \tag{4.1}$$

In the case of $L2$ regularization with regularization parameter λ, the loss is modified to read

$$\ell = \|\mathbf{Y} - \mathbf{X}\mathbf{P}_{XY}\|^2 + \lambda\|\mathbf{P}_{XY}\|^2. \tag{4.2}$$

The minimum of the regularized loss with respect to $\mathbf{P}_{XY}$ yields the solution $\mathbf{P}_{XY} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{Y}$.

4.1.2 Kernel Ridge Regression

Kernel ridge regression (KRR) [104, 132] is an extension of linear ridge regression to the RKHS and has been used successfully to build models for the prediction of atomic-scale properties [4–9, 12, 27, 77]. Substituting the RKHS feature vectors $\mathbf{\Phi}$ for $\mathbf{X}$ in Eqn. 4.2 gives the regularized loss

$$\ell = \|\mathbf{Y} - \mathbf{\Phi}\mathbf{P}_{\Phi Y}\|^2 + \lambda\|\mathbf{P}_{\Phi Y}\|^2, \tag{4.3}$$

so that the optimal weights are

$$\begin{aligned}\mathbf{P}_{\Phi Y} &= \left(\mathbf{\Phi}^T\mathbf{\Phi} + \lambda\mathbf{I}\right)^{-1}\mathbf{\Phi}^T\mathbf{Y} \\ &= \mathbf{\Phi}^T\left(\mathbf{\Phi}\mathbf{\Phi}^T + \lambda\mathbf{I}\right)^{-1}\mathbf{Y},\end{aligned} \tag{4.4}$$

where we have used Eqn. 20 of Ref. [133]. Predicted properties $\hat{\mathbf{Y}}$ can then be evaluated from the RKHS features, i.e., $\hat{\mathbf{Y}} = \mathbf{\Phi}\mathbf{P}_{\Phi Y}$. By using again the fact that $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T$, the regression weights can instead be expressed in terms of the kernel matrix, $\mathbf{P}_{KY} = \left(\mathbf{\Phi}\mathbf{\Phi}^T + \lambda\mathbf{I}\right)^{-1}\mathbf{Y} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{Y}$,

so that $\mathbf{P}_{\Phi Y} = \mathbf{\Phi}^T \mathbf{P}_{KY}$ [72]. The predicted targets can the be written equivalently as

$$\hat{\mathbf{Y}} = \mathbf{\Phi}\mathbf{\Phi}^T \mathbf{P}_{KY} = \mathbf{K}\mathbf{P}_{KY}. \tag{4.5}$$

4.1.3 Low-Rank Kernel Ridge Regression

A low-rank approximation to the kernel ridge regression solution can be achieved through the application of the Nyström approximation [104, 108, 134]. To construct the low-rank solution, we proceed as in standard KRR, but construct the loss using the Nyström approximation of the RKHS features $\mathbf{\Phi}_{NM}$,

$$\ell = \|\mathbf{Y} - \mathbf{\Phi}_{NM}\mathbf{P}_{\Phi Y}\|^2 + \lambda \|\mathbf{P}_{\Phi Y}\|^2, \tag{4.6}$$

for which the solution is

$$\begin{aligned} \mathbf{P}_{\Phi Y} &= \left(\mathbf{\Phi}_{NM}^T \mathbf{\Phi}_{NM} + \lambda \mathbf{I}\right)^{-1} \mathbf{\Phi}_{NM}^T \mathbf{Y} \\ &= \left(\mathbf{\Phi}_{NM}^T \mathbf{\Phi}_{NM} + \lambda \mathbf{I}\right)^{-1} \mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2} \mathbf{U}_{\mathbf{K}_{MM}}^T \mathbf{K}_{NM}^T \mathbf{Y}. \end{aligned} \tag{4.7}$$

Once again, we can define a set of weights that are instead based on the partial kernel matrix $\mathbf{K}_{NM}$,

$$\hat{\mathbf{Y}} = \mathbf{\Phi}_{NM}\mathbf{P}_{\Phi Y} = \mathbf{K}_{NM}\mathbf{U}_{\mathbf{K}_{MM}}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\mathbf{P}_{\Phi Y} = \mathbf{K}_{NM}\mathbf{P}_{KY}, \tag{4.8}$$

from which we see that

$$\begin{aligned} \mathbf{P}_{KY} &= \mathbf{U}_{\mathbf{K}_{MM}}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\mathbf{P}_{\Phi Y} \\ &= \mathbf{U}_{\mathbf{K}_{MM}}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\left(\mathbf{\Phi}_{NM}^T \mathbf{\Phi}_{NM} + \lambda \mathbf{I}\right)^{-1} \mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\mathbf{U}_{\mathbf{K}_{MM}}^T \mathbf{K}_{NM}^T \mathbf{Y}. \end{aligned} \tag{4.9}$$

By writing out explicitly $\mathbf{\Phi}_{NM}^T \mathbf{\Phi}_{NM}$ in terms of $\mathbf{K}_{NM}$ we obtain [104]

$$\mathbf{P}_{KY} = \left(\mathbf{K}_{NM}^T \mathbf{K}_{NM} + \lambda \mathbf{K}_{MM}\right)^{-1} \mathbf{K}_{NM}^T \mathbf{Y}. \tag{4.10}$$

4.2 Classification

Classification techniques are typically used to assign discrete labels to a set of samples. While a wide variety of classification algorithms exist, in the following we focus on the support vector classifier, as it serves as the primary means of classification in Chapters 6 and 7.

4.2.1 Support Vector Machines

Support vector machines (SVMs) are a class of supervised methods that share many similarities with perceptrons, the precusor to neural networks [128]. While SVMs can be used for both classification and regression [135], the use of SVMs in this thesis is restricted to classification problems. Hence, all future discussions of SVMs will refer exclusively to support vector classification.

The goal of support vector classification is to classify points in the feature space by dividing the feature space with a hyperplane $\mathbf{x}^T\mathbf{w} + b = 0$. Points on the positive side of the

hyperplane ($\mathbf{x}^T\mathbf{w} + b > 0$) are given the designated label "+1", while points on the negative side of the hyperplane ($\mathbf{x}^T\mathbf{w} + b < 0$) are labelled "−1". The "easiest" such binary classification problems are those cases in which the classes are linearly separable—that is, we are able to classify perfectly the data by dividing the feature space with a single hyperplane. If the data is linearly separable, however, it is often the case that there are multiple hyperplanes that can perfectly separate the classes. In cases where there are many possible hyperplanes, how do we know which one to choose? We can address this issue by introducing the concept of the *margin*, which grants the hyperplane a "thickness". By aiming to maximize the margin surrounding the hyperplane while forbidding any samples from falling within the margin, we consequently place the hyperplane at a location that maximizes the distances between the hyperplane and the nearest samples. Such a placement ensures that the resulting classification will generalize well to unseen samples [128]. This results in the optimization [73, 74],

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 \tag{4.11a}$$

$$\text{subject to} \qquad y_i(\mathbf{x}_i^T\mathbf{w} + b) \geq 1 \qquad \forall i = 1, \ldots, N. \tag{4.11b}$$

However, linearly separable classification problems are rather rare in practical situations, and in these cases it will be impossible to define a nonzero margin that does not contain any samples. In order to accommodate situations where the data are not linearly separable, or in problems that are linearly separable but we wish to allow for misclassifications to better account for the presence of outliers, we can introduce *slack variables* $\xi_i \geq 0$ and modify the optimization problem of Eqn. 4.11 to read [73, 74],

$$\min_{\mathbf{w},\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \tag{4.12a}$$

$$\text{subject to} \qquad y_i(\mathbf{x}_i^T\mathbf{w} + b) \geq 1 - \xi_i \qquad \text{and} \qquad \xi_i \geq 0 \qquad \forall i = 1, \ldots, N. \tag{4.12b}$$

This is known as the *soft margin* SVM, where the regularization parameter $C > 0$ scales the penalties imposed by the slack variables ξ_i, which serve as a measure for how strongly a sample is misclassified. For samples that are correctly classified and lie outside the margin, $\xi_i = 0$. Otherwise, ξ_i is proportional to the distance to margin boundary on the correct side of the hyperplane, i.e., $\xi_i = |y_i - (\mathbf{x}_i^T\mathbf{w} + b)|$. For $\xi_i > 1$, the sample is misclassified, and for $0 < \xi_i \leq 1$, the sample is correctly classified, but lies inside the margin and is still penalized [72, 73]. Given this construction, we can more intuitively write Eqn. 4.12 as [72]

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \max\left[0, 1 - y_i(\mathbf{x}_i^T\mathbf{w} + b)\right]. \tag{4.13}$$

However, as Eqn. 4.13 is non-differentiable, it is typically easier to solve the optimization problem of Eqn. 4.12 by minimizing the Lagrangian [73, 74],

$$\mathscr{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i - \sum_i \alpha_i(y_i(\mathbf{x}_i^T\mathbf{w} + b) - 1 + \xi_i) - \sum_i \mu_i\xi_i, \tag{4.14}$$

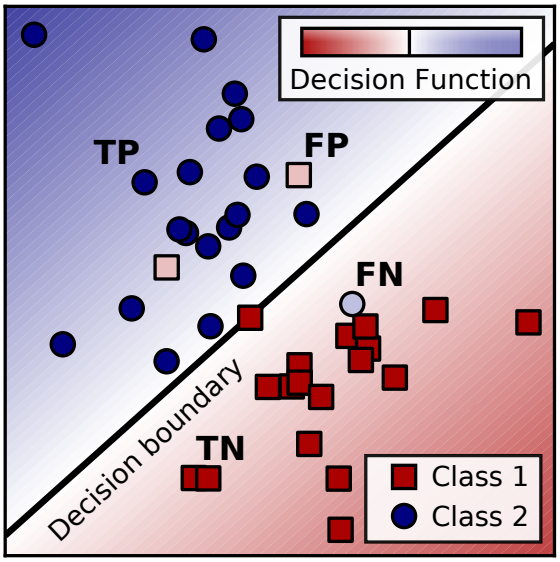

Figure 4.1 – Schematic of an SVM on toy data, showing the decision boundary and select points representing true positive (TP), false positive (FP), true negative (TN) and false negative (FN) predictions. The background is colored according to the value of the decision function, and misclassified points are shown with desaturated colors.

yielding $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$, $\sum_i \alpha_i y_i = 0$, and $\alpha_i = C - \mu_i$, where $\alpha_i \geq 0$ and $\mu_i \geq 0$ are Lagrange multipliers. Through the Karush-Kuhn-Tucker conditions, the Lagrangian can be converted to the *dual* form [73, 74],

$$\tilde{\mathscr{L}}(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \tag{4.15}$$

and maximized with the constraints $0 \leq \alpha_i \leq C$ and $\sum_i \alpha_i y_i = 0$ to obtain $\mathbf{w}$ and b. The coefficients α_i are only nonzero for points lying on the margin, which are known as the *support vectors*. Once $\mathbf{w}$ and b are determined, the *decision function* (or *confidence score*) $f(\mathbf{x}_i) = \mathbf{x}_i^T \mathbf{w} + b$ for a sample $\mathbf{x}_i$ can be computed and a class assignment made based on the sign of $f(\mathbf{x}_i)$. The dual formulation also makes obvious how the SVM can be extended to kernel methods. If the SVM is constructed in an RKHS, the dot product in Eqn. 4.15 is simply replaced with a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ [72–74]. A schematic of a linear SVM on toy data is provided in Fig. 4.1.

In the soft margin formulation, C is typically set using cross-validation. To understand the effect of C on the margin, recall that even correctly classified points will be penalized if they fall within the margin, as they have $\xi_i > 0$. Therefore, if C is large and the penalty for violating the margin is high, the width of the margin will shrink to avoid penalties on correctly classified examples. Conversely, if C is small and the penalty for violating the margin is low, the α_i are constrained to small values and the margin will be large.

While SVMs are powerful classification tools, they can generalize poorly if the there is a strong imbalance in the class labels y_i corresponding to the samples $\mathbf{x}_i$ used to construct the

model. In such cases, it is possible to define a class-specific penalty C_k that is applied to all samples in to class k [136], so that samples belonging to the minority class(es) can be assigned larger misclassification penalties.

Finally, we note that the discussion of SVMs above has been limited to binary classification problems; to perform multi-class classification, where the samples $\mathbf{x}_i$ can possess one of m labels, one can construct an ensemble of binary classifiers according to the "one vs. one" or "one vs. rest" schemes. In the "one vs. one" approach, $m(m-1)/2$ binary classifiers are constructed, one for each unique pair of classes. The final class assignment for a sample is then made based on the class that receives the most "votes" out of the $m(m-1)/2$ classifiers (though the case in which multiple classes are tied for the most votes is ambiguous). In the "one vs. rest" (also called "one vs. all") approach, only m binary classifiers are constructed, each one attempting to correctly distinguish the samples from class m from all other samples. The final class assignment for a sample $\mathbf{x}_i$ is made based on the class with the largest corresponding decision function value $f(\mathbf{x}_i)$ [73]. While the "one vs. one" approach requires a large number of classifiers, each binary classifier involves only a subset of the samples. Conversely, the "one vs. rest" approach trains fewer binary classifiers but each classifier must be trained on all of the samples.

5 Combined Learning [1]

While the standalone unsupervised and supervised algorithms presented in Chapters 3 and 4 can be applied to great effect in informatics applications, a deeper understanding of the available data can be acquired through combined, or hybrid, techniques that unify the supervised and unsupervised learning paradigms. Such a unification can be achieved either through algorithmic machinery that leverages both the regressors and regressor targets, or through workflows that integrate supervised and unsupervised learning into a single pipeline, where the outputs of one method are used as the inputs to another. This chapter will address both approaches.

To begin, an overview of principal covariates regression (PCovR) [137–142] is provided, serving as our primary example of a method that combines supervised and unsupervised learning into a single algorithm; similar combined learning techniques include (kernel) partial least squares [143–146] and (kernel) continuum regression [147, 148]. The central idea underlying all of these methods is to generate a low-dimensional latent space from which we can make regression predictions; the main distinguishing features of these approaches are in the exact details of how the latent space is constructed.

This chapter concludes with a brief discussion of how supervised and unsupervised learning can be integrated into a single workflow to provide additional insight about relationships within the data and to facilitate their interpretation.

5.1 Principal Covariates Regression

The aim of PCovR [137] is to construct a low-dimensional latent space that simultaneously minimizes the information loss incurred by projecting a set of samples $\mathbf{X}$ into the low-dimensional space and the error in predicting the target properties $\mathbf{Y}$ from the latent space representation $\mathbf{T}$. This task is achieved by minimizing a weighted sum of the PCA and linear regression losses (Eqns. 3.9 and 4.1), where the mixing parameter α is used to assign the relative importance of

[1] Sections 5.1–5.3 of this chapter are adapted with modifications under the Creative Commons Attribution 4.0 (CC BY 4.0) license from Helfrecht, B. A., Cersonsky, R. K., Fraux, G. & Ceriotti, M. Structure-Property Maps with Kernel Principal Covariates Regression. *Machine Learning: Science and Technology* **1,** 045021. doi:10.1088/2632-2153/aba9ef (2020); all authors contributed to the writing of the manuscript from which the present text has been adapted.

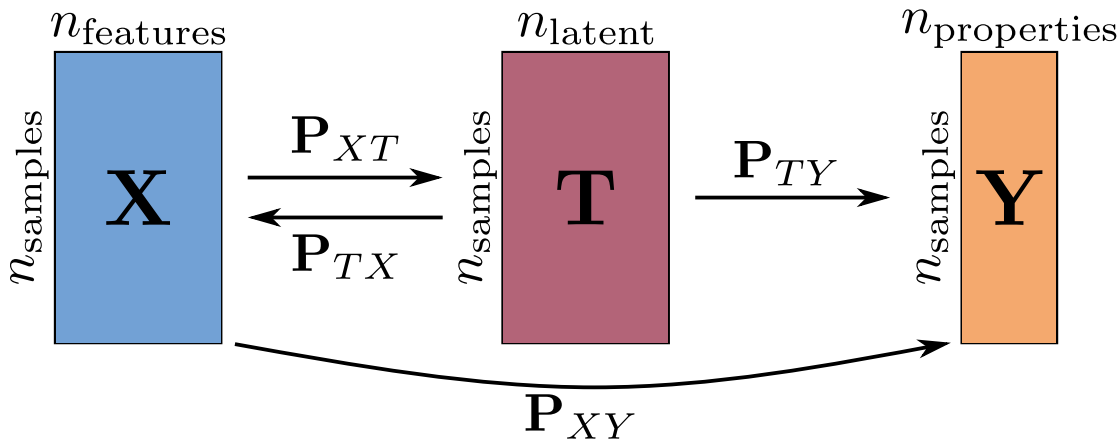

Figure 5.1 – Schematic of the PCovR notation, showing the transformations between the input features **X**, the latent space **T**, and the targets **Y** through the matrices **P**. Adapted from Ref. [107] under CC BY 4.0.

the individual loss terms,

$$\ell = \alpha\|\mathbf{X} - \mathbf{X}\mathbf{P}_{XT}\mathbf{P}_{TX}\|^2 + (1-\alpha)\|\mathbf{Y} - \mathbf{X}\mathbf{P}_{XT}\mathbf{P}_{TY}\|^2. \tag{5.1}$$

To ensure that the unweighted magnitudes of two terms of the loss function are approximately equal, we enforce that **X** and **Y** have been columnwise centered and subsequently scaled to have a Frobenius norm of 1. The derivation that follows is based closely on the original formulation of PCovR [137] with some notational differences that make transparent the relationship between PCovR and the methods introduced in Chapters 3 and 4, as well as the extensions described in Sections 5.2 and 5.3. A schematic of the notation scheme is presented in Figure 5.1.

PCovR can be formulated in one of two ways. The first, which we refer to as *sample-space* PCovR, involves the diagonalization of a modified Gram matrix of the feature vectors and is preferable when the number of features n_{features} is larger than the number of samples n_{samples}. The second, which we call *feature-space* PCovR, is more suitable when $n_{\text{samples}} > n_{\text{features}}$, and involves the construction and diagonalization of a modified covariance of the input features. Both approaches yield the same latent-space projections and regression predictions.

Sample-space PCovR

Minimizing the PCovR loss in Eqn. 5.1 is most straightforward when we impose orthonormality in the latent space and endeavor to find projections $\tilde{\mathbf{T}} = \mathbf{X}\mathbf{P}_{X\tilde{T}}$ such that $\tilde{\mathbf{T}}^T\tilde{\mathbf{T}} = \mathbf{I}$. If we define $\mathbf{P}_{\tilde{T}X} = \tilde{\mathbf{T}}^T\mathbf{X}$ and $\mathbf{P}_{\tilde{T}Y} = \tilde{\mathbf{T}}^T\mathbf{Y}$, we can rewrite the loss as

$$\ell = \alpha\|\mathbf{X} - \tilde{\mathbf{T}}\tilde{\mathbf{T}}^T\mathbf{X}\|^2 + (1-\alpha)\|\mathbf{Y} - \tilde{\mathbf{T}}\tilde{\mathbf{T}}^T\mathbf{Y}\|^2. \tag{5.2}$$

Rather than directly minimizing this loss, we can equivalently maximize the similarity

$$\rho = \mathrm{Tr}\left(\alpha\tilde{\mathbf{T}}\tilde{\mathbf{T}}^T\mathbf{X}\mathbf{X}^T + (1-\alpha)\tilde{\mathbf{T}}\tilde{\mathbf{T}}^T\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T\right) \tag{5.3}$$

$$= \mathrm{Tr}\left(\alpha\tilde{\mathbf{T}}\tilde{\mathbf{T}}^T\mathbf{X}\mathbf{X}^T + (1-\alpha)\tilde{\mathbf{T}}\tilde{\mathbf{T}}^T\mathbf{X}\mathbf{P}_{XY}\mathbf{P}_{XY}^T\mathbf{X}^T\right), \tag{5.4}$$

where we have substituted $\mathbf{Y}$ with the regression approximation $\hat{\mathbf{Y}} = \mathbf{X}\mathbf{P}_{XY} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$; the similarity is invariant to this substitution as a result of the definitions of $\hat{\mathbf{Y}}$, $\tilde{\mathbf{T}}$, and the cyclic properties of the trace. If we define the modified Gram matrix as

$$\tilde{\mathbf{K}} = \alpha\mathbf{X}\mathbf{X}^T + (1-\alpha)\mathbf{X}\mathbf{P}_{XY}\mathbf{P}_{XY}^T\mathbf{X}^T, \tag{5.5}$$

we can more compactly write the similarity as

$$\rho = \mathrm{Tr}\left(\tilde{\mathbf{T}}^T\tilde{\mathbf{K}}\tilde{\mathbf{T}}\right), \tag{5.6}$$

from which we can see that the similarity is maximized when the latent space projections $\tilde{\mathbf{T}}$ are the principal eigenvectors of the matrix $\tilde{\mathbf{K}}$, i.e., $\tilde{\mathbf{T}} = \hat{\mathbf{U}}_{\tilde{\mathbf{K}}}$. We can additionally define a set of latent space projections $\mathbf{T}$ that become consistent with those of linear-kernel KPCA (and thus standard PCA and classical MDS) as $\alpha \to 1$ by multiplying the orthogonal latent space projections $\tilde{\mathbf{T}}$ by the corresponding square roots of the eigenvalues of $\tilde{\mathbf{K}}$ to give $\mathbf{T} = \tilde{\mathbf{T}}\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{K}}}^{1/2} = \tilde{\mathbf{K}}\hat{\mathbf{U}}_{\tilde{\mathbf{K}}}\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{K}}}^{-1/2}$. The matrix $\mathbf{P}_{XT}$ that transforms the input features $\mathbf{X}$ to the latent space representation $\mathbf{T}$ is thus

$$\mathbf{P}_{XT} = \left(\alpha\mathbf{X}^T + (1-\alpha)\mathbf{P}_{XY}\mathbf{P}_{XY}^T\mathbf{X}^T\right)\hat{\mathbf{U}}_{\tilde{\mathbf{K}}}\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{K}}}^{-1/2}, \tag{5.7}$$

and we can similarly define a matrix $\mathbf{P}_{TY}$ corresponding to the regression weights for the prediction of the targets $\mathbf{Y}$ from the latent space representation $\mathbf{T}$,

$$\mathbf{P}_{TY} = \left(\mathbf{T}^T\mathbf{T} + \lambda\mathbf{I}\right)^{-1}\mathbf{T}^T\mathbf{Y} \underset{\lambda\to 0}{=} \hat{\mathbf{\Lambda}}_{\tilde{\mathbf{K}}}^{-1/2}\hat{\mathbf{U}}_{\tilde{\mathbf{K}}}^T\mathbf{Y}. \tag{5.8}$$

We can also construct the matrix $\mathbf{P}_{TX}$ that reconstructs the original features from the latent space by regressing instead on the original features, yielding

$$\mathbf{P}_{TX} = \hat{\mathbf{\Lambda}}_{\tilde{\mathbf{K}}}^{-1/2}\hat{\mathbf{U}}_{\tilde{\mathbf{K}}}^T\mathbf{X}. \tag{5.9}$$

As $\alpha \to 0$, the regression weights $\mathbf{P}_{XY} = \mathbf{P}_{XT}\mathbf{P}_{TY}$ are those of the pure linear regression solution in $\mathbf{X}$ as long as the number of latent space components is greater than or equal to the number of columns of $\mathbf{Y}$.

Feature-space PCovR

The optimal PCovR projections can also be determined by diagonalizing a modified covariance matrix $\tilde{\mathbf{C}}$ in place of the modified Gram matrix in Eqn. 5.5. Given that $\mathbf{I} = \tilde{\mathbf{T}}^T\tilde{\mathbf{T}} = \mathbf{P}_{X\tilde{T}}^T\mathbf{X}^T\mathbf{X}\mathbf{P}_{X\tilde{T}} = \mathbf{P}_{X\tilde{T}}^T\mathbf{C}\mathbf{P}_{X\tilde{T}}$, we can see that $\mathbf{C}^{1/2}\mathbf{P}_{X\tilde{T}}$ is orthogonal and thus rewrite the similarity function from

Eqn. 5.6 as

$$\rho = \mathrm{Tr}\left(\mathbf{P}_{X\tilde{T}}^T \mathbf{C}^{1/2} \tilde{\mathbf{C}} \mathbf{C}^{1/2} \mathbf{P}_{X\tilde{T}}\right), \tag{5.10}$$

introducing

$$\begin{aligned} \tilde{\mathbf{C}} &= \mathbf{C}^{-1/2} \mathbf{X}^T \tilde{\mathbf{K}} \mathbf{X} \mathbf{C}^{-1/2} \\ &= \alpha \mathbf{C} + (1-\alpha) \mathbf{C}^{-1/2} \mathbf{X}^T \hat{\mathbf{Y}} \hat{\mathbf{Y}}^T \mathbf{X} \mathbf{C}^{-1/2}. \end{aligned} \tag{5.11}$$

The similarity is maximised when the orthogonal matrix $\mathbf{C}^{1/2}\mathbf{P}_{X\tilde{T}}$ matches the principal eigenvectors $\hat{\mathbf{U}}_{\tilde{\mathbf{C}}}$ of $\tilde{\mathbf{C}}$, i.e. $\mathbf{P}_{X\tilde{T}} = \mathbf{C}^{-1/2}\hat{\mathbf{U}}_{\tilde{\mathbf{C}}}$. In general $\mathbf{P}_{X\tilde{T}}\mathbf{P}_{\tilde{T}X} = \mathbf{C}^{-1/2}\hat{\mathbf{U}}_{\tilde{\mathbf{C}}}\hat{\mathbf{U}}_{\tilde{\mathbf{C}}}^T\mathbf{C}^{1/2}$ is not a symmetric matrix, and so it is not possible to define an orthogonal $\mathbf{P}_{XT}$ such that $\mathbf{P}_{TX} = \mathbf{P}_{XT}^T$. Similarly to sample space PCovR, we can obtain projections $\mathbf{T}$ that reduce to those of PCA, linear KPCA, and classical MDS from the orthogonal projections $\tilde{\mathbf{T}}$ by multiplying by the square root of the eigenvalues of $\tilde{\mathbf{C}}$, so that

$$\mathbf{P}_{XT} = \mathbf{P}_{X\tilde{T}}\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{C}}}^{1/2} = \mathbf{C}^{-1/2}\hat{\mathbf{U}}_{\tilde{\mathbf{C}}}\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{C}}}^{1/2}, \tag{5.12}$$

from which we can determine the matrices $\mathbf{P}_{TX}$ and $\mathbf{P}_{TY}$,

$$\mathbf{P}_{TX} = \hat{\mathbf{\Lambda}}_{\tilde{\mathbf{C}}}^{-1/2}\hat{\mathbf{U}}_{\tilde{\mathbf{C}}}^T\mathbf{C}^{1/2} \tag{5.13}$$

$$\mathbf{P}_{TY} = \hat{\mathbf{\Lambda}}_{\tilde{\mathbf{C}}}^{-1/2}\hat{\mathbf{U}}_{\tilde{\mathbf{C}}}^T\mathbf{C}^{-1/2}\mathbf{X}^T\mathbf{Y}. \tag{5.14}$$

5.2 Kernel Principal Covariates Regression

Principal covariates regression can be extended to nonlinear analyses through the use of kernel methods. This can be achieved by substituting $\mathbf{\Phi}$ for $\mathbf{X}$ (and being centered and scaled likewise) to construct the augmented kernel matrix

$$\tilde{\mathbf{K}} = \alpha\mathbf{K} + (1-\alpha)\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T, \tag{5.15}$$

which is similar to Eqn. 5.5 except that $\mathbf{K}$ is the kernel matrix corresponding to the RKHS features $\mathbf{\Phi}$ and $\hat{\mathbf{Y}} = \mathbf{K}(\mathbf{K}+\lambda\mathbf{I})^{-1}\mathbf{Y}$ is the kernel ridge regression solution for $\mathbf{Y}$. A schematic similar to that of Fig. 5.1 is provided for the KPCovR notation in Fig. 5.2. Just as in PCovR, the unit variance projections $\tilde{\mathbf{T}}$ are given by the principal eigenvectors $\hat{\mathbf{U}}_{\tilde{\mathbf{K}}}$ of $\tilde{\mathbf{K}}$, with the modified projections $\mathbf{T} = \tilde{\mathbf{T}}\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{K}}}^{1/2}$ defined accordingly. The projections $\mathbf{T}$ thus approximate the features $\tilde{\mathbf{\Phi}}$ for the RKHS corresponding to the modified kernel $\tilde{\mathbf{K}} = \tilde{\mathbf{\Phi}}\tilde{\mathbf{\Phi}}^T$.

Given that we typically wish to avoid explicit computation of the RKHS features, we can determine the latent space representations by performing the projection directly from the kernel matrix such that $\mathbf{T} = \mathbf{K}\mathbf{P}_{KT}$,

$$\mathbf{P}_{KT} = \left(\alpha\mathbf{I} + (1-\alpha)\left(\mathbf{K}+\lambda\mathbf{I}\right)^{-1}\mathbf{Y}\hat{\mathbf{Y}}^T\right)\hat{\mathbf{U}}_{\tilde{\mathbf{K}}}\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{K}}}^{-1/2}. \tag{5.16}$$

We determine the matrix $\mathbf{P}_{TY}$ that enables predictions of properties from the KPCovR latent

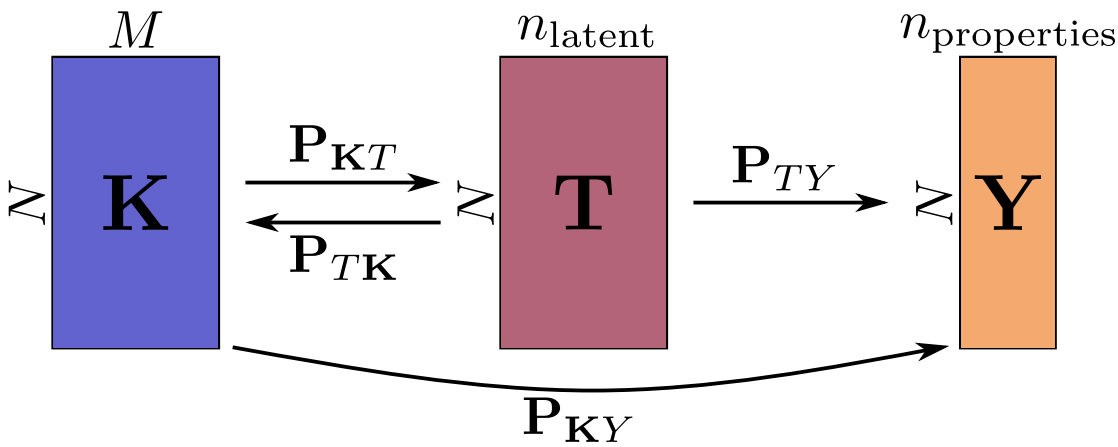

Figure 5.2 – Schematic of the KPCovR notation, showing the transformations between the input kernel matrix **K**, the latent space **T**, and the targets **Y** through the matrices **P**. Adapted from Ref. [107] under CC BY 4.0.

space **T** just as in the linear case (Eqn. 5.8). For completeness we can similarly define the matrices $\mathbf{P}_{TX}$ and $\mathbf{P}_{TK}$ by regressing from **T** onto **X** and **K**, but these transformations are of less practical utility.

5.3 Low-Rank Kernel Principal Covariates Regression

Just as it is possible to construct low-rank versions of KPCA and KRR, one can also use the Nyström approximation to derive a low-rank version of KPCovR. This is accomplished by performing feature-space PCovR on the Nyström approximations to the RKHS features $\mathbf{\Phi}_{NM} = \mathbf{K}_{NM}\mathbf{U}_{\mathbf{K}_{MM}}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}$, so that the covariance reads

$$\begin{aligned}\mathbf{C} &= \mathbf{\Phi}_{NM}^T\mathbf{\Phi}_{NM} && (5.17)\\ &= \mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\mathbf{U}_{\mathbf{K}_{MM}}^T\mathbf{K}_{NM}^T\mathbf{K}_{NM}\mathbf{U}_{\mathbf{K}_{MM}}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}, && (5.18)\end{aligned}$$

and can be used to define the modified KPCovR covariance

$$\begin{aligned}\tilde{\mathbf{C}} &= \alpha\mathbf{C}\\ &+ (1-\alpha)\mathbf{C}^{1/2}\left(\mathbf{C}+\lambda\mathbf{I}\right)^{-1}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\mathbf{U}_{\mathbf{K}_{MM}}^T\mathbf{K}_{NM}^T\mathbf{Y}\\ &\times\mathbf{Y}^T\mathbf{K}_{NM}\mathbf{U}_{\mathbf{K}_{MM}}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\left(\mathbf{C}+\lambda\mathbf{I}\right)^{-1}\mathbf{C}^{1/2}.\end{aligned} \tag{5.19}$$

We can thus define a matrix $\mathbf{P}_{\Phi T}$ analogous to Eqn. 5.12,

$$\mathbf{P}_{\Phi T} = \mathbf{C}^{-1/2}\hat{\mathbf{U}}_{\tilde{\mathbf{C}}}\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{C}}}^{1/2}, \tag{5.20}$$

through which we can define the matrix $\mathbf{P}_{KT}$,

$$\mathbf{P}_{KT} = \mathbf{U}_{\mathbf{K}_{MM}}\mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2}\mathbf{C}^{-1/2}\hat{\mathbf{U}}_{\tilde{\mathbf{C}}}\hat{\mathbf{\Lambda}}_{\tilde{\mathbf{C}}}^{1/2}, \tag{5.21}$$

and the matrix $\mathbf{P}_{TY}$, noting the similarity with Eqn. 5.14 and removing the explicit dependence on the approximate RKHS features,

$$\mathbf{P}_{TY} = \hat{\mathbf{\Lambda}}_{\tilde{\mathbf{C}}}^{-1/2} \hat{\mathbf{U}}_{\tilde{\mathbf{C}}}^{T} \mathbf{C}^{-1/2} \mathbf{\Lambda}_{\mathbf{K}_{MM}}^{-1/2} \mathbf{U}_{\mathbf{K}_{MM}}^{T} \mathbf{K}_{NM}^{T} \mathbf{Y}. \tag{5.22}$$

The inverse transformation matrices $\mathbf{P}_{TK}$ and $\mathbf{P}_{TX}$ can again be defined through regressions from the latent space projections $\mathbf{T}$.

5.4 Generalized Convex Hull

It is a commonly used tactic to predict the thermodynamic stability of a multiphase material or chemical system through a convex hull construction [149], where the most stable phase, or mixture of phases, is that with the lowest Gibbs free energy. In these constructions, the Gibbs free energy is usually represented as a function of the fractional composition of the system.

In materials discovery efforts, however, one is typically more interested in the relative stability of many different materials rather than multiple phases of the same material. In such cases, we can construct a convex hull that indicates which materials are the most thermodynamically stable by abstracting the composition indicator to a general structural descriptor. This was the approach taken by Anelli et al. in Ref. [150] in their *generalized convex hull* (GCH) construction. The idea of the GCH is to iteratively build a convex hull in an abstract feature space that reflects the probability that a particular structure lies on the convex hull, taking into account any uncertainties in the computed energies and atomic structures.

At each iteration of the GCH construction, the error σ_{G_k} in the (free) energy G_k relative to the convex hull for a structure k is estimated as [150]

$$\sigma_{G_k} = \epsilon \sqrt{\sigma_G^{-2} \sum_i g_i (\mathbf{x}_k - \sum_{\mathbf{x}_j \in \mathscr{H}} w_{kj} \mathbf{x}_j)^2}, \tag{5.23}$$

where ϵ is the error in computed energies from a set of reference values for the dataset, $\mathscr{H}$ the set of structures that lie on the current guess of the convex hull, σ_G^2 the variance of the structure energies, w_{kj} the coefficients describing the piecewise linear segments of the hull as a mixture of adjacent vertices, and g_i a function yielding the (ridge) regression predictions of the energies based on changes in the features $\mathbf{x}_k$, which are typically taken to be a transformation of the raw feature vectors through, e.g., KPCA. Similarly, uncertainties in the atomic structures are calculated from small random displacements of the individual atoms in a number of reference configurations [150],

$$\sigma_{\mathbf{x}_k} = \sqrt{\frac{1}{n_s n_r} \sum_r^{n_r} \sum_s^{n_s} (\mathbf{x}_r^{(s)} - \mathbf{x}_r)^2}, \tag{5.24}$$

where n_r and n_s are the number of reference configurations and the number of randomizations for each reference configuration r. $\mathbf{x}_r$ denotes the features corresponding to the configuration r, and $\mathbf{x}_r^{(s)}$ the features of the sth randomization of the configuration r. Once the uncertainties are obtained, the structural features and energies of each structure k are

perturbed by the amounts σ_{G_k} and $\sigma_{\mathbf{x}_k}$, and a new convex hull is constructed. Those configurations that continue to appear on or near the convex hull after many successive error estimations and perturbations are judged as having a higher probability of being stable. Given these probabilistic measures of stability, the GCH procedure can be repeated multiple times, gradually pruning away those structures with the lowest probabilities of being vertices until a predetermined minimum probability threshold is surpassed [150]. The remaining configurations are taken to be the vertices of the GCH.

5.5 Sequential Workflows

In addition to algorithms that combine supervised and unsupervised learning through a unified loss function, such as PCovR-based methods, it is also sometimes useful to combine supervised and unsupervised learning through sequential workflows. Perhaps the simplest of these workflows are those employed by methods such as (kernel) principal components regression [151–156] and clusterwise regression [157].

In principal components regression, one simply performs a regression analysis on the principal components of the original regressor variables, and a subset or all of the principal components may be retained for the regression. One typically wishes to retain the components that are are most predictive of the targets or that are associated with the greatest regression weight [158] which may or may not be the components associated with the greatest variance [151]. The idea of applying principal component analysis to the predictor variables and using the resulting components in other learning methods is quite general, as it can be used as a preprocessing step for other algorithms.

The idea behind clusterwise regression is also rather simple: one partitions the feature space through some clustering algorithm, and then applies linear regression independently to each of the clusters. Thus for k clusters, k different regression models are constructed, each tailored for a particular cluster. Like principal components regression, clusterwise regression can be combined with other methods, including PCovR [159].

The general idea behind these kinds of workflows is to apply some learning algorithm, supervised or unsupervised, on a set of data and pass the outputs to a different algorithm, where they serve as the model inputs, additionally permitting the construction of specialized workflows designed for a particular use case. This approach is used in Chapter 7, where the decision functions from a support vector machine are used to define a PCovR-based feature space on which a convex hull is constructed for the purpose of identifying synthesis candidates from a large database of hypothetical zeolite frameworks.

6 Structural Motifs in Proteins [1]

6.1 Introduction

Understanding structure–property relationships in complex materials is often tied to the understanding of local atomic motifs. This is particularly true for biopolymers, which exhibit motifs at different length scales and whose shape largely determines their interaction with other molecules, with proteins serving as the archetypal example. Interactions between residues in the protein backbone give rise to a sequence of *secondary structures*, such as α-helices and β-sheets, which further assemble into tertiary and quaternary structures. These multiscale assemblies are primarily determined by hydrogen bonding and backbone dihedral angle patterns, which can be considered as the "building blocks" of protein structure. Understanding the folding and assembly of biopolymers and predicting the structure of novel macromolecules thus requires good knowledge of the hydrogen bonding and dihedral angle motifs in these materials, including their structural characteristics, likelihood of occurring, and how they interact to form larger patterns.

Consequently, much work has been dedicated to understanding and classifying hydrogen bonds, ultimately producing several geometric and energetic criteria to identify their presence or absence [161–170]. Likewise, examining the patterns of the backbone dihedral angles in a macromolecule has found widespread use in chemistry, biology, and biophysics to aid in the identification of protein secondary structure [171], often through the use of the Ramachandran plot [172] to visualize the distribution of dihedral angles.

Such motif-based rationales have been successfully employed to identify the secondary structure of proteins: the DSSP [173] and STRIDE [171] algorithms are two notable examples. However, the identification of structural motifs in proteins is often based on a combination of domain knowledge, human intuition, and—sometimes generous—approximations, and may not be unique or readily applicable to different classes of macromolecules. Moreover, motif definitions are typically based on assessments of specific structures or, in the case of the hydrogen bond, focus only on a single subset of the atomic species that may be involved.

[1] This chapter is adapted with modifications under the Creative Commons Attribution 4.0 (CC BY 4.0) license from Helfrecht, B. A., Gasparotto, P., Giberti, F. & Ceriotti, M. Atomic Motif Recognition in (Bio)Polymers: Benchmarks From the Protein Data Bank. *Frontiers in Molecular Biosciences* **6,** 24. doi:10.3389/fmolb.2019.00024 (2019); BAH performed the data analysis and prepared figures, PG ran preliminary tests, and all authors contributed to the design of the study and to the writing of the manuscript from which the present text has been adapted.

In this context, a statistical framework that is capable of automatically identifying structural motifs and applicable to multiple domains without relying on heuristics would be advantageous. A purely data-driven definition of various motifs would be particularly useful in the field of bioinformatics, where such motifs are used for structure prediction or in the development of scoring functions for processes like protein-ligand docking. For example, Rosetta, one of the most well-known energy functions, has been developed to predict the structure of a protein given its amino acid sequence and local structural features such as dihedral angles [174, 175].

Another situation where purely data-driven motif definitions would be advantageous is in secondary structure classification. While several methods have been developed to classify protein secondary structures [171, 173, 176–182], these methods tend to rely on amino acid sequences, hydrogen bonding energies, geometrical criteria, or some combination thereof. Machine learning techniques [183], and neural networks in particular [180, 181, 184–190] have also been used to classify protein secondary structures based on a variety of features. Other schemes have been developed to classify conformational patterns and secondary structure using dihedral angles alone [182, 191], but there remains a lack of a truly agnostic method for classifying (and predicting) secondary structures.

In this chapter, we demonstrate the utility of unsupervised machine learning in constructing a statistical definition of atomic-scale motifs. Given a descriptor of the atomic environments, we construct a probability density of the feature space that is subsequently partitioned using the Probabilistic Analysis of Molecular Motifs (PAMM) algorithm [110, 111], which casts the probability density into a Gaussian mixture model (GMM), which we can use to find the most probable motifs in the distribution. We construct the density distribution using two different feature representations: one based on classical geometric descriptors such as interatomic distances and dihedral angles, and another, more agnostic scheme that uses the SOAP representation [84, 85, 192]. The motif "fingerprints" obtained through the partitioning of the feature space probability density have a general definition and are transferable between different systems. To illustrate this point, rather than selecting proteins of a given family or with small variations in the sequence, we construct the data-driven motifs based on structures from the Research Collaboratory for Structural Bioinformatics Protein Data Bank (RCSB PDB) [64]. We compare the PAMM-based motifs to more "traditional" geometric and energetic definitions of the hydrogen bond and to DSSP- and STRIDE-assigned secondary structures to assess their similarity. We then use supervised learning to understand whether the differences in secondary structure assignments are due to the identification of the motifs themselves or as a result of a lack of descriptive power in the feature space.

6.2 Hydrogen Bonding Motifs

As a first benchmark of the application of automatic pattern recognition schemes to (bio)polymers, we consider the case of the hydrogen bond (HB), for which we construct a data-driven definition based on the PAMM scheme [110, 111] to identify modes in the probability density corresponding to atomic patterns. As discussed in Section 3.1.1, the PAMM algorithm takes as input a feature representation for which it constructs a kernel density estimation on a sparse

grid obtained by subsampling the input data. A density-based clustering is then performed to identify local maxima in the estimate of the probability density. Each identified cluster is represented as a Gaussian mode, making it possible to define probabilistic motif identifiers (PMIs), structural indicators, or "fingerprints", that take a value between zero and one and represent the degree of confidence by which a new local structure can be assigned to each of the clusters. While there is no shortage of geometry-based HB definitions, and PAMM has already been applied to the identification of HBs in water and ammonia [110, 193], the exercise of identifying HBs in proteins offers a chemically diverse test case where there exist concrete, domain-specific definitions for comparison.

6.2.1 Hydrogen Bond Data Selection

All of the structures used in the definition of the structural motifs, regardless of the underlying descriptor used, were obtained from the RCSB PDB database on January 31, 2018 among those for which experimental data is available. Note that the PDB contains redundant entries, i.e., protein structures with very similar sequences. These redundant structures were included in our analyses, and so the resulting models are biased according to the redundancies of the PDB. The downside of using experimentally determined structures as the basis of our analysis is that the structural precision—particularly for hydrogen atoms—is limited and varies greatly between PDB entries. Given that hydrogen positions are obviously central to the definition of a hydrogen bonding motif, we included in our analysis only those protein crystal structures obtained by X-ray diffraction with a resolution better than 1.2 Å where hydrogen atom positions were available. Only 872 structures in the PDB met these requirements and could be properly parsed. Given that each structure contains hundreds of hydrogen bonds, this amount of data proved sufficient for our statistical analysis. From each protein structure, we examined four different hydrogen bond flavors: (1) N–H···N, (2) N–H···O, (3) O–H···O, and (4) O–H···N, considering only N, O, and H atoms with occupancy ≥ 0.95. Any oxygen and hydrogen atoms belonging to water or other small molecules were excluded.

6.2.2 Geometry Descriptors

For the determination of hydrogen bonding motifs, we examined all triplets of atoms, for which one atom (O or N) is considered as the putative donor, one atom (O or N) is considered as the putative acceptor, and the third atom is the H atom taking part in the bond. We did not use any additional criterion to identify which atoms could be part of a hydrogen bond, which means that the analysis considers as putative hydrogen bonds also triplets in which the three atoms are chemically bound or adjacent to one another in the backbone or in a side chain. Most of the traditional definitions of hydrogen bonds would implicitly discard these configurations; however, in the spirit of reducing the amount of domain-specific knowledge implicit in the motif definitions, we have retained them to serve as a demonstration of the robustness of a statistical, unsupervised approach for identifying distinct structural patterns.

Even in protein structures obtained from high-resolution X-ray diffraction, hydrogen positions are often "refined". In other words, each hydrogen atom is often fixed at a predetermined distance from the atom to which it is covalently bound [194, 195]. To ensure

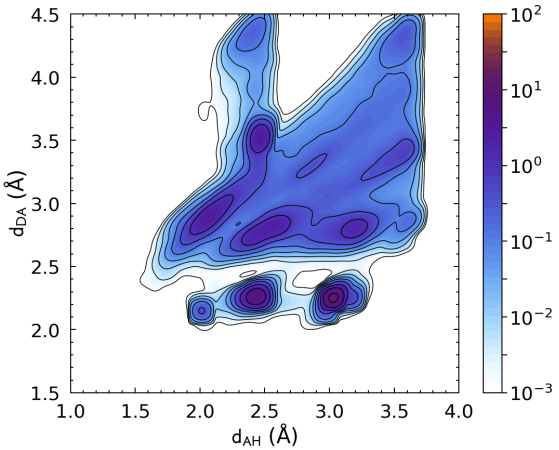

Figure 6.1 – Total probability density of d_{AH} and d_{DA} across all hydrogen bond flavors. The distribution is peaked strongly at (d_{AH} = 3.0, d_{DA} = 2.25) as a result of common N–H···O geometries in the protein backbone corresponding to N and O atoms in the same or directly adjacent residues. Contours are equally spaced on a logarithmic scale. Reproduced from Ref. [160] under CC BY 4.0.

that this artificial feature would not further bias the clustering, only the donor–acceptor and acceptor–hydrogen distances were chosen as geometrical descriptors for each hydrogen bond. Ignoring the donor–hydrogen distance does not limit the resolving power of a PAMM analysis, but makes it impossible to automatically eliminate some configurations with very large donor–hydrogen distances. For this reason, before proceeding with the clustering, we further filtered the hydrogen bonds using the same geometric criteria that has been used in earlier studies of hydrogen bonding in water [110, 193], which relies on all of the donor–acceptor, donor–hydrogen, and acceptor–hydrogen distances (d_{DA}, d_{DH}, and d_{AH}, respectively). Those triplets for which $d_{DH} + d_{AH} > 4.5$ Å were discarded in addition to those in which $d_{DH} > d_{AH}$. The latter refinement reduces redundancies when examining different hydrogen bond flavors, as a given triplet with $d_{DH} > d_{AH}$ in N–H···O is equivalent to that same triplet with $d_{DH} < d_{AH}$ in O–H···N; the donor and acceptor labels have just been interchanged. With these refinements, we identified 418,865 N–H···N triplets, 918,014 N–H···O triplets, 42,650 O–H···O triplets, and 57,572 O–H···N triplets that were subsequently used to build the Gaussian mixture models. The probability density of acceptor–hydrogen and donor–acceptor distances of the 1,437,101 donor–hydrogen–acceptor triplets across all four hydrogen bond flavors is shown in Fig. 6.1.

6.2.3 Clustering Parameters

To reduce the computational cost of the clustering procedure while ensuring adequate coverage of the (d_{AH}, d_{DA}) feature space, we selected a sparse grid of 2,000 representative configurations on which we computed a kernel density estimation of the probability distribution of different motifs. The representative configurations were selected using FPS [106, 123]. The KDE bandwidth and local scale factors were determined automatically as discussed in Ref. [111]; the automatically determined bandwidth was scaled by a factor of 0.3 to account for the

strong multi-modality of the distribution, while we found the automatic choice of quick-shift distance to be appropriate. Clusters with weights less than 10^{-5} in the resulting mixture model were discarded, as they were sparsely populated and did not meaningfully contribute to the overall probability distribution and could be considered outliers.

6.2.4 Probabalistic Motif Indentifiers (PMIs)

For each hydrogen bond flavor, the PMI $f(\mathbf{x})$ at a point $\mathbf{x} = (d_{AH}, d_{DA})$ is calculated as outlined in Section 3.1.1 [110, 111],

$$f(\mathbf{x}) = \frac{p_{HB} G(\mathbf{x}; \boldsymbol{\mu}_{HB}, \boldsymbol{\Sigma}_{HB})}{P(\mathbf{x}) + \zeta}, \tag{6.1}$$

where p_{HB} is the weight of the Gaussian G with mean $\boldsymbol{\mu}_{HB}$ and covariance $\boldsymbol{\Sigma}_{HB}$ describing the cluster corresponding to the hydrogen bond, ζ is the background parameter, set to 10^{-5} for our purposes, and $P(\mathbf{x})$ is the total probability density of the GMM,

$$P(\mathbf{x}) = \sum_{k}^{N} p_k G(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{6.2}$$

where N is the total number of clusters in the model.

For comparison, we used the following as the definition for the PMI of a distance–angle geometry-based definition of the hydrogen bond:

$$f(\mathbf{x}) = \begin{cases} 1, & d_{DA} < 3.5\,\text{Å and } d_{AH} < 2.5\,\text{Å and } d_{DH} < 1.5\,\text{Å and } \angle ADH < 30.0^\circ \\ 0, & \text{else.} \end{cases} \tag{6.3}$$

As another example, the DSSP [173] definition of an N–H···O hydrogen bond, which is based on the distances d between the atoms participating in the C=O bond of one residue and the N–H bond of another residue, can also be used to construct a PMI. To construct the DSSP-based PMI, we computed the required DSSP distances for all {N, H, C, O} quadruplets in each protein for which all four atoms have occupancy ≥ 0.95, and mapped the quadruplet to the (d_{AH}, d_{DA}) space simply by taking d_{AH} as the oxygen–hydrogen distance and d_{DA} as the nitrogen–oxygen distance. The DSSP hydrogen bonding dataset was based on the same 872 protein crystal structures used for our other HB analyses, but only 844 of these contained valid (N–H, C=O) pairs according to the criteria outlined in Sections 6.2.1 and 6.2.2. Hence, the DSSP hydrogen bonding dataset included 552,281 potential N–H···O hydrogen bonds.

For each $\mathbf{x} = (d_{AH}, d_{DA})$, we computed DSSP HB PMI based on the joint probability distribution

$$P_{HB}(\mathbf{x}) = P(\mathbf{x}, E_{DSSP} < -0.5\ \text{kcal/mol}), \tag{6.4}$$

where E_{DSSP} is the DSSP electrostatic energy as defined in Ref. [173],

$$E_{DSSP} = q_1 q_2 f \left(\frac{1}{d_{ON}} + \frac{1}{d_{CH}} - \frac{1}{d_{OH}} - \frac{1}{d_{CN}} \right), \tag{6.5}$$

where the factors $q_1 = 0.42e$, $q_2 = 0.20e$, and $f = 332$ gives the energy E in kcal/mol with d in angstroms and e as the unit electron charge. Configurations with $E_{DSSP} < -0.5$ kcal/mol are considered by DSSP to be hydrogen bonds [173].

The DSSP-based PMI can then be constructed following Eqns. 6.1 and 6.2 by replacing $G(\mathbf{x}; \boldsymbol{\mu}_{HB}, \boldsymbol{\Sigma}_{HB})$ with the joint probability density $P_{HB}(\mathbf{x})$ and by defining the total probability density as

$$P(\mathbf{x}) = p_{HB} P_{HB}(\mathbf{x}) + (1 - p_{HB}) P(\mathbf{x}, E_{DSSP} \geq -0.5 \text{ kcal/mol}). \tag{6.6}$$

where the weight p_{HB} is the fraction of (C=O, N−H) pairs that have $E < -0.5$ kcal/mol.

In order to compare different HB definitions and to quantify how often they disagree in identifying a local motif in the feature space as an HB, we introduce the quantity

$$\delta_{AB} = \frac{1}{\lambda} \frac{\int P_{total}(\mathbf{x}) f_A(\mathbf{x}) f_B(\mathbf{x}) \mathrm{d}\mathbf{x}}{\int P_{total}(\mathbf{x}) \left[f_A(\mathbf{x}) + f_B(\mathbf{x}) - f_A(\mathbf{x}) f_B(\mathbf{x}) \right] \mathrm{d}\mathbf{x}}, \tag{6.7}$$

which is the probability that the PMIs A and B both identify the point $\mathbf{x} = (d_{AH}, d_{DA})$ as an HB relative to the probability that either one or the other identify $\mathbf{x}$ as an HB. $P_{total}(\mathbf{x})$ is the total probability distribution of observing (d_{AH}, d_{DA}) in the PDB dataset across all hydrogen bond flavors. The numerator is thus the expected number of hydrogen bonds in the dataset that are common to both A and B, and the denominator is the total expected number of hydrogen bonds in the dataset (points $\mathbf{x}$ that are classified as hydrogen bonds by A only, B only, or both A and B). The normalization factor λ is included to account for the fact that the PMIs f are posterior probabilities rather than true probability distributions. Thus, λ is chosen such that Eqn. 6.7 is equal to one when $f_A(\mathbf{x}) = f_B(\mathbf{x})$,

$$\lambda = \sqrt{\frac{\int P_{total}(\mathbf{x}) f_A^2(\mathbf{x}) \mathrm{d}\mathbf{x}}{\int P_{total}(\mathbf{x}) \left[2 f_A(\mathbf{x}) - f_A^2(\mathbf{x}) \right] \mathrm{d}\mathbf{x}} \cdot \frac{\int P_{total}(\mathbf{x}) f_B^2(\mathbf{x}) \mathrm{d}\mathbf{x}}{\int P_{total}(\mathbf{x}) \left[2 f_B(\mathbf{x}) - f_B^2(\mathbf{x}) \right] \mathrm{d}\mathbf{x}}}. \tag{6.8}$$

6.2.5 Analysis of PMIs

Having outlined a data-driven, unsupervised definition of the hydrogen through the PMI and a means of comparing different hydrogen bond definitions, we can begin to make assessments of the relative merits of using unsupervised machine learning for identifying structural motifs in complex materials. Serving as a benchmark for the analysis is the traditional distance–angle hydrogen bond definition, the PMI of which is shown in Fig. 6.2(a) as the highlighted area superimposed upon a contour plot of the the (d_{AH}, d_{DA}) probability distribution including all four HB flavors. The PMI for the distance–angle definition encompasses a large peak in $P(\mathbf{x})$ that indeed corresponds to hydrogen-bonded configurations, but it also includes a few additional peaks. By inspection, we found that these additional modes are associated with motifs in which the putative donor and acceptor atoms are part of the same amino acid residue or where the H atom is not chemically bound to the donor. In practice, these geometries would be discarded a priori because any practical application of the distance–angle definition would likely take covalent bonding information into account; the specious geometries could

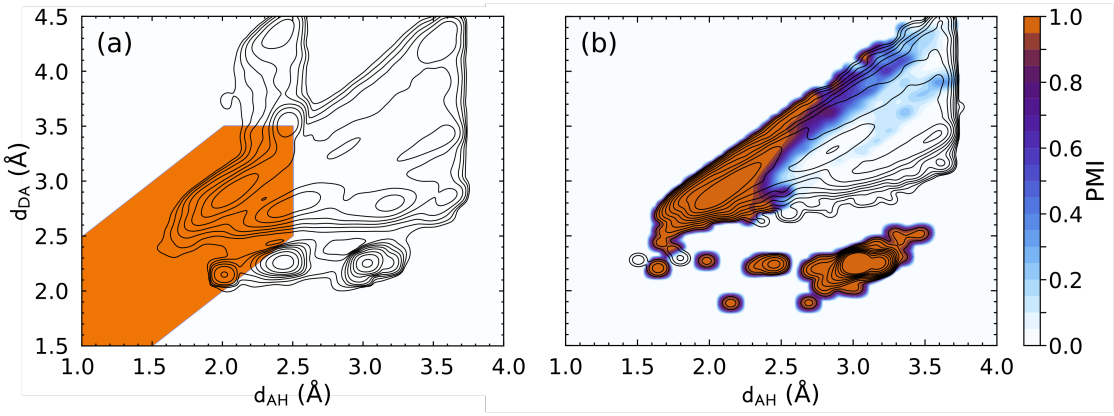

Figure 6.2 – (a) Histogram of the acceptor–hydrogen and donor–acceptor distances across all hydrogen bond flavors, plotted with log-spaced contours. The maximum at ($d_{AH} \approx 2.1$ Å, $d_{DA} \approx 2.8$ Å) corresponds to the typical H-bond range. Other maxima are associated with other structural features, such as covalently bound groups on the side chains, geometries in which the two electronegative atoms are in the same residue, or configurations in which the hydrogen atom is not bound to the donor. The orange-shaded area corresponds to the distance-angle PMI as defined in Eqn. 6.3. (b) Density plot of the PMI constructed using the DSSP hydrogen bond definition with $\zeta = 10^{-5}$. The PMI is plotted on top of a histogram of the distance features for N–H···O hydrogen bonds (discarding non-backbone groups, and any triplet for which it is not possible to define a DSSP H-bond energy, e.g. due to partial occupations), with log-spaced contours. DSSP identifies very clearly the H-bond peak, but also picks up specious correlations corresponding to residues that are immediately adjacent to one another (peak at ($d_{AH} \approx 3.0, d_{DA} \approx 2.25$)). Adapted from Ref. [160] under CC BY 4.0.

also be excluded by manually modifying the cutoff distances and angles in the PMI definition, but this approach requires knowledge of the underlying probability distribution and thus lacks transferability. Figure 6.2(a) thus underscores the complex heuristics and domain-specific knowledge that is often necessary when using even well-established definitions to recognize atomic-scale motifs, and serves as a warning of the risks one could incur when blindly following these prescriptions in a different context than originally intended, e.g., where assumptions of fixed chemical connectivity no longer hold.

Similar considerations apply to the DSSP definition for N–H···O HBs, whose corresponding PMI is shown in Fig. 6.2(b). The DSSP definition follows more closely the main HB peak of the distribution, as one would expect given that it is heavily fine-tuned for N–H···O bonds between peptide groups. At the same time, DSSP also requires further heuristics to discard specious correlations corresponding to N–H and C=O in immediately adjacent residues, where ($d_{AH} \approx 3.0, d_{DA} \approx 2.25$).

Contrast the distance–angle and DSSP HB PMIs to the top row of Fig. 6.3, which shows the PAMM PMIs for each cluster in the GMMs, computed separately for each HB flavor. The four distributions differ substantially from each other, and from the overall $P(\mathbf{x})$ (Fig. 6.1), while exhibiting multiple modes that are correctly identified by PAMM and assigned different cluster indices. Some of these modes correspond to correlations between covalently bound

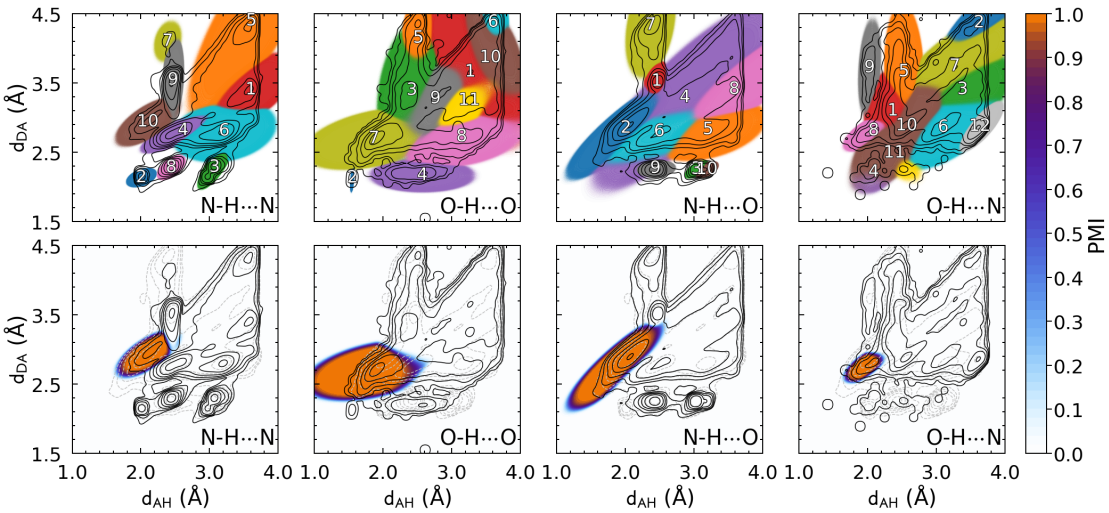

Figure 6.3 – The top panels represent all the clusters identified by PAMM for each HB flavor. The clusters are numbered in an arbitrary order, and the colors reflect the cluster that is dominant in each region, as determined by its corresponding PMI (as defined in Eqn. 6.1, computed with $\zeta = 10^{-5}$). The bottom panels highlight the PMI of the cluster associated with the hydrogen bond. Reproduced from Ref. [160] under CC BY 4.0.

atoms, while others correspond to longer-range correlations. For each flavor, the cluster that corresponds to the hydrogen bond is that with its center (mode) nearest to ($d_{AH} = 1.82$ Å, $d_{DA} = 2.74$ Å) [110]. The corresponding PMIs, which are plotted in the bottom row of Fig. 6.3, identify with great precision the region in the probability distribution that corresponds to the HB, and eliminate automatically the specious configurations due to adjacent residues or covalently bound groups without the need for additional heuristics.

A PAMM clustering of backbone-only N–H···N and N–H···O triplets is shown in Fig. 6.4. No hydrogen bond cluster is evident in the N–H···N case, suggesting that the N–H···N hydrogen bonds we observe occur almost exclusively between amino acid side chains. Similarly, the shape and location of the backbone-only N–H···O hydrogen bond PMI is very similar to that of the total N–H···O hydrogen bond PMI, suggesting that the N–H···O hydrogen bonds are predominantly those existing in the protein backbone. Note that the distribution for N–H···O configurations in the backbone (Fig. 6.4) is different from the distribution in the DSSP definition (Fig. 6.2(b)), which also considers backbone N–H···O geometries. This is because the DSSP definition applies additional constraints on the types of "acceptable" geometries, namely that the N and O atoms must be in different residues and that the H atom must be bound to the N atom, again illustrating the importance of domain-specific knowledge in heuristic-based definitions and their resulting lack of transferability.

Fig. 6.3 also shows that different HB flavors correspond to noticeably different regions of the (d_{AH}, d_{DA}) feature space. This suggests that a substantial fraction of molecular patterns would be misclassified if one tried to transfer the HB definition from one flavor to another. As shown in Table 6.1, the probability that two definitions yield the same classification, as measured by Eqn. 6.7, can be as low at 50%. Fig. 6.6 provides a visualization of the over-

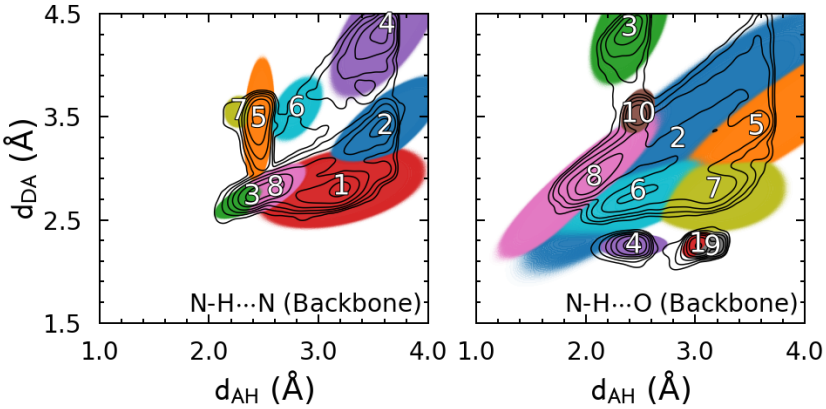

Figure 6.4 – PAMM clustering for N–H···O and N–H···N backbone geometries with a background parameter $\zeta = 10^{-5}$, where the donor and acceptor atoms are a part of the protein backbone only. Reproduced from Ref. [160] under CC BY 4.0.

lap between the PMIs of the different HB flavors. The agreement between the data-driven PMIs and the conventional distance–angle definition is even poorer, as shown in Table 6.2 and in Fig. 6.5. It should be stressed, however, that this is largely due to the inclusion of correlations that are usually discarded by additional heuristics: if one computes the PMI similarity using a probability distribution $P_{total}(\mathbf{x})$ that discards atoms in the same or nearby residues, the probability increases substantially, particularly for N–H···N and N–H···O, as these are the flavors responsible for the majority of specious hydrogen bond geometries (e.g., intra-arganine or intra-histidine N–H···N triplets and backbone N–H···O triplets with donor and acceptor atoms in directly adjacent residues). The increase in PMI similarity is generally less pronounced when comparing two different hydrogen bond flavors because these PMIs are derived from a PAMM GMM, which automatically recognizes the specious geometries as separate motifs. This example, although simple, demonstrates how one can use data-analytic techniques to extract definitions of molecular motifs based on experimental structural data. It also serves as a reminder of how heuristic definitions can lack transferability, and how their apparent simplicity is often contingent on a considerable amount of prior knowledge and the enforcement of additional conditions.

6.3 Secondary Structure Motifs

Having compared automatic, unsupervised motif definitions against more "traditional" definitions for the case of the hydrogen bond in proteins, we apply a similar analysis for the case of backbone dihedral angle and secondary structure patterns, for which the variety and complexity of motifs is much greater. Secondary structure patterns play a central role in rationalizing the structure and behavior of proteins, and there exist well-established definitions based on the identification of HBs along the protein backbone, such as STRIDE [171] and DSSP [173]. There is, however, a need for definitions of secondary structure that are based on continuous structural coordinates, for instance, to bias atomistic simulations or to perform structure searches [196, 197]. As an example of how one can use an automatic, unsupervised scheme

Table 6.1 – Probabilities that two PMIs corresponding to different hydrogen bond flavors agree that a point **x** is a hydrogen bond (Eqn. 6.7). The superscripts (i) and $(i+1)$ correspond to probabilities δ_{AB} where $P_{total}(\mathbf{x})$ excludes donor–hydrogen–acceptor triplets in which the donor and acceptor atoms are in the same residue (i), or additionally in adjacent residues $(i+1)$.

PMI A	PMI B	δ_{AB}	$\delta_{AB}^{(i)}$	$\delta_{AB}^{(i+1)}$
N–H···N	N–H···O	0.92	0.93	0.94
N–H···N	O–H···O	0.57	0.63	0.74
N–H···N	O–H···N	0.60	0.59	0.60
O–H···O	N–H···O	0.55	0.61	0.71
O–H···O	O–H···N	0.60	0.68	0.85
N–H···O	O–H···N	0.57	0.57	0.58

Table 6.2 – Probabilities that the hydrogen bond PMI and the distance–angle definition agree that a point **x** is a hydrogen bond (Eqn. 6.7). The superscripts (i) and $(i+1)$ correspond to probabilities δ_{AB} where $P_{total}(\mathbf{x})$ excludes donor–hydrogen–acceptor triplets in which the donor and acceptor atoms are in the same residue (i), or additionally in directly adjacent residues $(i+1)$.

Bond Type	δ_{AB}	$\delta_{AB}^{(i)}$	$\delta_{AB}^{(i+1)}$
N–H···N	0.56	0.65	0.89
N–H···O	0.60	0.71	0.93
O–H···O	0.63	0.65	0.68
O–H···N	0.33	0.39	0.53

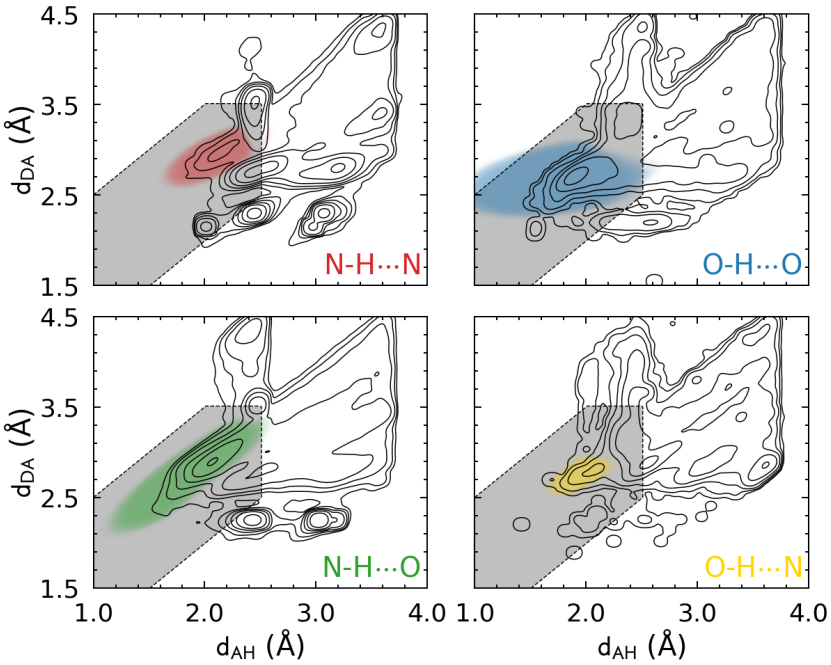

Figure 6.5 – Comparison between the PAMM PMIs of the four hydrogen bond flavors and the distance–angle hydrogen bond definition superimposed on a histogram of the acceptor–hydrogen and donor–acceptor distances for the hydrogen bond flavor of interest. Contours are equally spaced on a logarithmic scale. Reproduced from Ref. [160] under CC BY 4.0.

such as PAMM to provide a definition of secondary structure motifs, we compare PAMM-based motifs against the DSSP and STRIDE secondary structure definitions [2]. To this end, we constructed PMIs based on two different feature representations. The first representation is based on the Ramachandran dihedrals [172], which provide a simple, local description of the protein backbone and whose correlation to secondary structure has been long appreciated [189, 191, 198]. The second representation, based on the SOAP descriptor, provides a more detailed (though abstract) description of the atomic environment surrounding each backbone residue while requiring minimal chemical intuition and being transferable to different systems, as the only required information is the positions of the atoms in the protein backbone.

6.3.1 Dihedral Angle Representation

Because the calculation of dihedral angles ϕ and ψ is not sensitive to hydrogen atomic positions, the PAMM analysis of dihedral angles included all experimental protein crystal structures from the RCSB PDB (as of January 31, 2018) obtained from X-ray diffraction with a resolution better than 1.5 Å, totaling 12,708 structures and 4,275,677 residues from which dihedral angles could be extracted using Biopython [199]. Note again that no measures were taken to discard redundant structures from the PAMM analysis, hence the resulting mixture model is biased according to the redundancies of the PDB. In addition to the two-dimensional

[2] The DSSP motifs were calculated using version 2.2.1 of the software; no version information was available for STRIDE

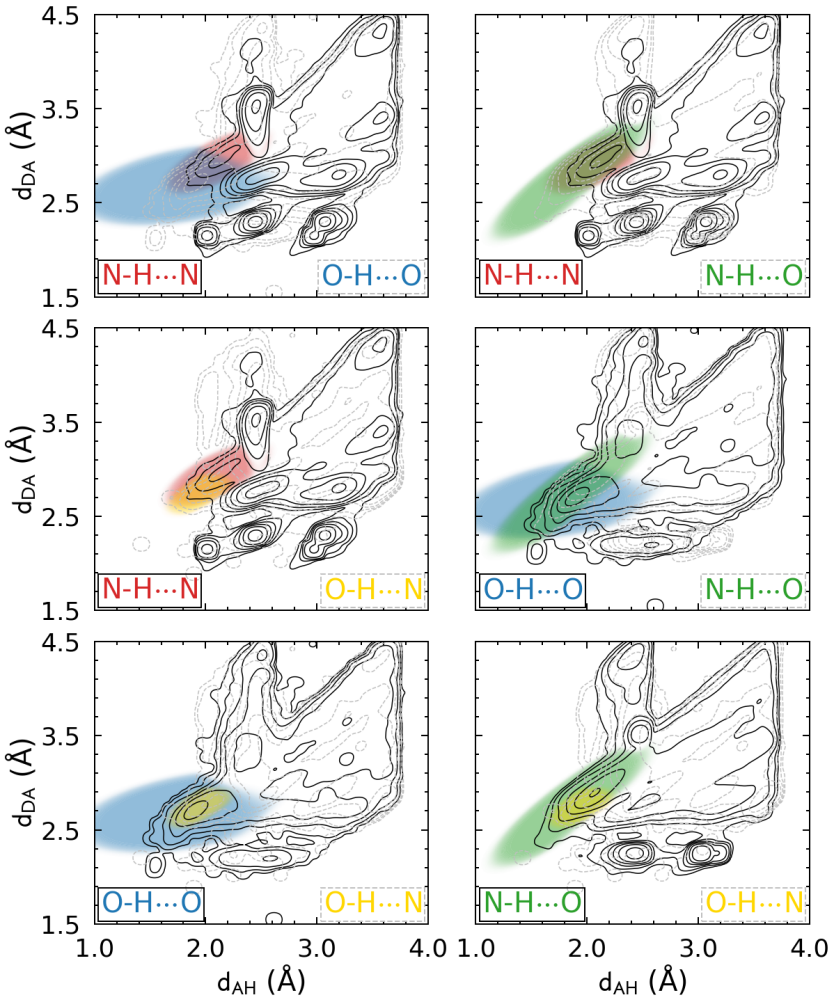

Figure 6.6 – Comparison between the PAMM PMIs of the four different hydrogen bond flavors. The linestyle of the box enclosing the label of the hydrogen bond flavor corresponds to the linestyle of the log-spaced contours of the underlying (d_{AH}, d_{DA}) distribution for that hydrogen bond flavor. Reproduced from Ref. [160] under CC BY 4.0.

(ϕ, ψ) representation, we constructed representations based on chains of ϕ and ψ angles in three and five consecutive residues, resulting in six- and ten-dimensional feature spaces.

6.3.2 SOAP Representation

The same 12,708 structures used to build the dataset of dihedral angles were also used for the SOAP-based feature representation [3]. Although SOAP is a powerful descriptor, the high dimensionality of the SOAP vectors makes PAMM pattern recognition based on these descriptors computationally intractable for large datasets. Therefore, we first performed a principal component analysis (PCA) of the SOAP vectors with the aim of reducing the dimension of the input space for PAMM while maintaining the most discriminating SOAP features of the individual proteins. To accelerate this process, we used an FPS subset of SOAP components to reduce the input space for the PCA while maintaining its span. In particular, we selected 100 random structures and computed the SOAP vectors for all of the C_α atoms in the selected structures, with the local environment comprising all C, N, and O atoms within a cutoff radius of 6.0 Å, which is large enough to incorporate information on several neighboring residues. From this collection of SOAP vectors, we selected 200 SOAP components via FPS, using the squared Euclidean distance between the SOAP vectors as the measure of separation [200]. The SOAP vectors centered around all C_α atoms were then computed for all structures just as they were for the random subset, but only the FPS components were retained and used to build the PCA representation; all other components of the SOAP vector were discarded. To match the dimensionality of the backbone-dihedral-based representations, we constructed separate representations including the first 2, 6 and 10 PCA components of the reduced SOAP vectors. The computation of the SOAP vectors was carried out using `quippy` [201], expanding the atomic density using 12 radial basis functions and 9 angular functions. The cutoff transition width and width for the atomic Gaussians was set to 0.5.

6.3.3 Clustering Parameters

Fingerprints of backbone dihedral angle motifs were computed using PAMM, where the underlying KDE was based on 4,000 sample points in the (ϕ, ψ) feature space selected with FPS. A scaling factor of 0.15 was applied to the KDE bandwidth, and a scaling of the quick-shift threshold of 0.20 was employed, as we found that the values determined automatically based on the heuristics discussed in Ref. [111] were smoothing excessively the distribution, resulting in a loss of resolving power. We determined the optimal parameters by monitoring the number of clusters and their robustness as assessed by a bootstrapping analysis. For the six- and ten-dimensional representations we again used 4,000 sample points for the KDE but selected a bandwidth scaling factor of 0.30 and set the quick-shift scaling to 0.80.

Fingerprints for the SOAP-based motifs were computed similarly, using 4,000 KDE grid

[3] The atomic positions of the proline of residue 2 in chain E of structure 3ADM are identical to the atomic positions of residue 5 of the same chain. Overlapping atomic positions causes the SOAP representation to fail, and so residue 2 of chain E in structue 3ADM was discarded (in addition to the nitrogen of residue 3, which has identical coordinates to the nitrogen of residue 6). Therefore, 4,275,676 residues were included in our SOAP analysis. These residues—common to both the dihedral angle and SOAP datasets—were used for the support vector machine computations of Q3 and Q8 scores, discussed in Section 6.3.6.

points and a quick shift parameter of 1.0. The KDE bandwidth was chosen to be $\sigma_i = f_s\sqrt{\mathrm{Tr}(\mathbf{\Sigma})}$, where $\mathbf{\Sigma}$ is the covariance of the data in the feature space and f_s is 0.20, 0.50, and 0.80 for the two-, six-, and ten-dimensional representations respectively. Similar to the case of the HB, we discarded clusters with weights less than 10^{-5} for both the dihedral angle and SOAP GMMs.

6.3.4 Analysis of PMIs

The PMIs for each of the Gaussians in a PAMM GMM of the dihedral angles ϕ and ψ are shown in Fig. 6.7. The PAMM dihedral angle clustering agrees well with those obtained by Hollingsworth et al. [191] and Nagy and Oostenbrink [182], who have previously developed classification schemes based solely on dihedral angles. However, we observe like Hollingsworth et al. that dihedral angle patterns do not necessarily correspond to established secondary structure definitions, which is made clear upon comparison of Fig. 6.8, which shows 100,000 randomly selected dihedral angle pairs colored according to their DSSP and STRIDE secondary structure assignments, and the clusters presented in Fig. 6.7. For reference, the DSSP and STRIDE secondary structure classifications are as follows: B, isolated β-bridge; E, extended strand; G, 3_{10}-helix; H, α-helix; I, π-helix; T, turn; S, bend (DSSP only); C, loop, irregular element, or none of the above ("coil"). We use an "X" to signify an amino acid residue for which no secondary structure was assigned.

As a first step towards understanding the lack of correspondence between the dihedral angle motifs and conventional secondary structure definitions, we also examined the cluster assignments in the six-dimensional feature space. To facilitate the visualization of this higher-dimensional feature space, we applied the Sketch-map dimensionality reduction method [121–123] using 500 landmark points and setting $\sigma = 2.5$, $a_X = b_X = 4$, and $a_T = b_T = 2$; the resulting projection is shown in in Fig. 6.9. The Sketch-map projection corroborates our earlier observations that, with the exception of the helices and strands, any given secondary structure is distributed widely across the high-dimensional space. As will be discussed in Section 6.3.6, the lack of correspondence between the dihedral angle motifs and established secondary-structure classifications is not due to an intrinsic lack of resolving power, but to the fact that dihedrals emphasize different kinds of structural correlations so that secondary structure motifs are not associated with separate modes of the feature space.

6.3.5 Probability Distributions

To quantify the agreement (or lack thereof) between the PMIs and conventional secondary structure definitions, we can use a framework based on the joint and conditional probability distributions of the PAMM cluster assignments and DSSP and STRIDE secondary structure classifications. Given that each point in the feature space $\mathbf{x}$ can be associated with a single amino acid residue, it can be paired with a DSSP or STRIDE secondary structure classification y and a PAMM cluster assignment A with probability $p^{(A)}(\mathbf{x})$. The joint probability distribution $P(A, y)$ can thus be constructed by summing the cluster probabilities over all points $\mathbf{x}_y$ with secondary structure y,

$$P(A, y) = \frac{1}{N}\sum_{\mathbf{x}_y} p^{(A)}(\mathbf{x}_y), \tag{6.9}$$

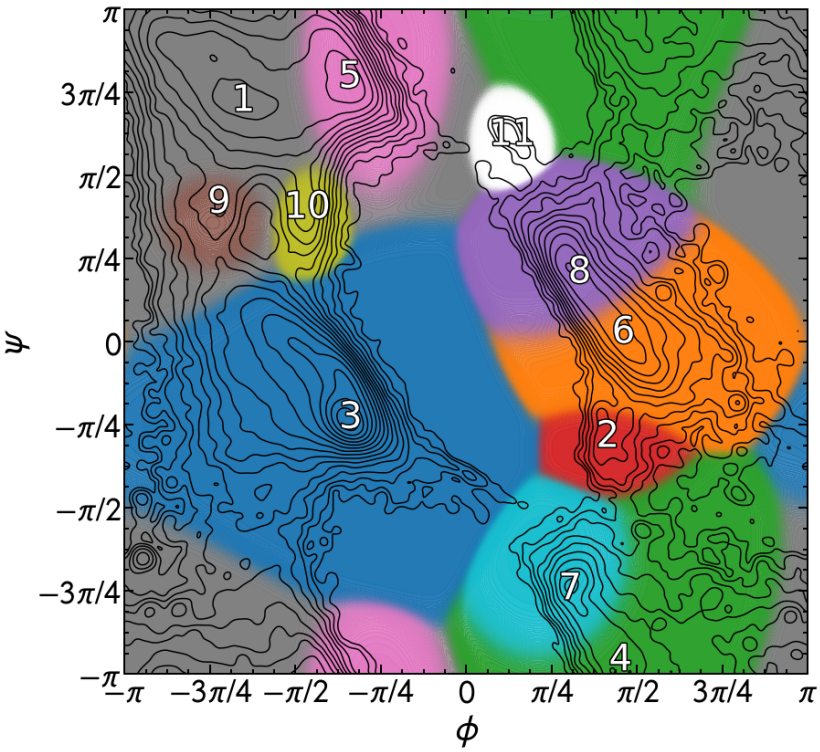

Figure 6.7 – PAMM clustering of all calculated dihedral angles with $\zeta = 0$. Cluster numbers are placed at the mode of the cluster, and each cluster has been colored differently. The isocontours of the total distribution are equally spaced on a logarithmic scale. Reproduced from Ref. [160] under CC BY 4.0.

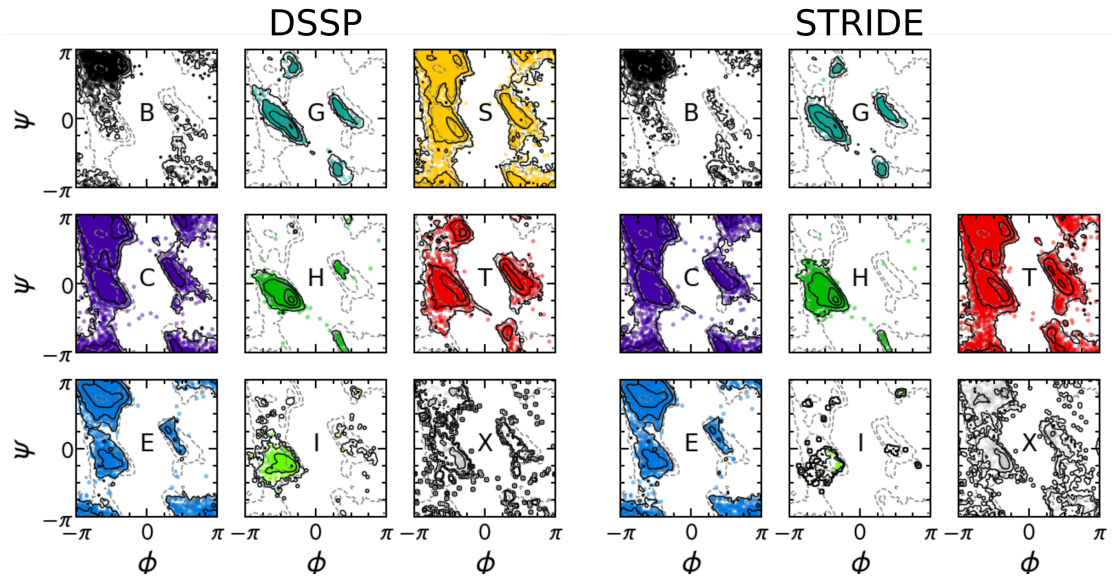

Figure 6.8 – Collection of 100,000 randomly selected (ϕ, ψ) pairs, separated according to the DSSP and STRIDE secondary structure classification of each pair. Solid contours correspond to the distribution of the secondary structure of interest; dashed contours correspond to the total distribution of all ϕ, ψ angles. Contours are equally spaced on a logarithmic scale. Adapted from Ref. [160] under CC BY 4.0.

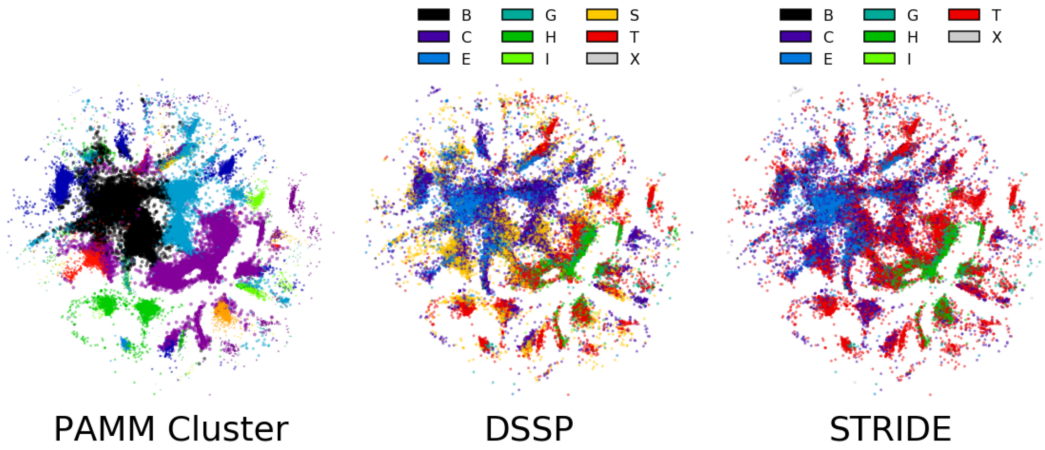

Figure 6.9 – Sketch-map representations of 100,000 randomly selected points in the six-dimensional ϕ, ψ space. Each point is colored according to its PAMM cluster assignment and middle residue DSSP or STRIDE secondary structure assignment. The lack of clear grouping observed among secondary structures suggests that secondary structure cannot be assigned based on dihedral angles alone. The points that are colored by their PAMM cluster are also sized based on the cluster weight; points belonging to a cluster with higher weight are larger. Adapted from Ref. [160] under CC BY 4.0.

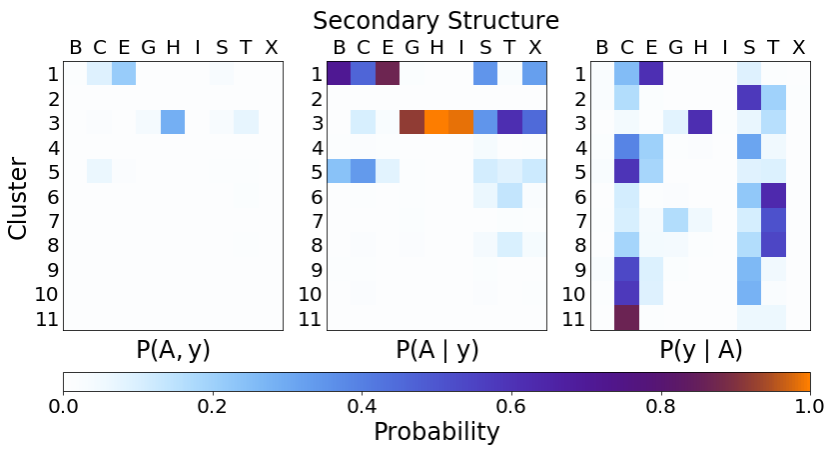

Figure 6.10 – Joint and conditional probabilities for the secondary structures obtained from DSSP and the clustering of dihedral angles from PAMM, where A is the cluster assignment and y the secondary structure classification. Reproduced from Ref. [160] under CC BY 4.0.

where N is the total number of residues considered. Based on the joint probability, we can compute the marginals $P(A)$ and $P(y)$ and the conditional probabilities $P(A \mid y)$ and $P(y \mid A)$, which provide equivalent information and make it easy to identify the correspondence—if any—between the PAMM-based PMI and the conventional definitions.

One can summarize the ability of the automatic definition to reproduce the classification given by STRIDE or DSSP by viewing the joint probability $P(A, y)$ in the framework of the Q3 (or Q8) accuracy score [186]. Given a particular clustering arrangement, one or more clusters can be selected that individually correspond to strands (B, E), helices (G, H, I) or coils (C, S, T) by assigning each cluster A the secondary structure that maximizes $P(y \mid A)$. Thus, for sets of clusters $\mathscr{E}, \mathscr{H}, \mathscr{C}$ corresponding to strands, helices, and coils, the Q3 score is the sum $Q_{\mathscr{E}} + Q_{\mathscr{H}} + Q_{\mathscr{C}}$, where

$$Q_{\mathscr{E}} = \sum_{i \in \mathscr{E}} (P(i, B) + P(i, E)) \tag{6.10a}$$

$$Q_{\mathscr{H}} = \sum_{j \in \mathscr{H}} \big(P(j, G) + P(j, H) + P(j, I)\big) \tag{6.10b}$$

$$Q_{\mathscr{C}} = \sum_{k \in \mathscr{C}} (P(k, C) + P(k, S) + P(k, T)) \,. \tag{6.10c}$$

Fig. 6.10 gives the joint and conditional probability distributions of the PAMM cluster assignment and the DSSP secondary structure assignment. (The probability distributions using the STRIDE secondary structure assignment are very similar to those using the DSSP assignment, and can be found in Appendix B.)

Fig. 6.10 shows that no one secondary structure (labelled by $y \in \{B, C, E, G, H, I, S, T, X\}$) is confined to a single PAMM cluster (labelled by $A \in \{1, \ldots, 11\}$), through the helices (G, H, I) and strands (B, E) are more strongly localized than the other secondary structures, with $A = 1, y = E$ and $A = 3, y = H$ being by large the most probable mutual assignments. The

joint probability distribution, however, is not easy to interpret because of the widely varying populations of the different clusters. For this reason, Fig. 6.10 also shows the conditional probabilities, which normalize the joint assignments based on the DSSP and PAMM marginals, yielding $P(A \mid y)$ and $P(y \mid A)$, respectively. The distribution conditional on DSSP assignments shows that a large fraction of E and H motifs are assigned to PAMM Clusters 1 and 3, while the distribution conditional on PAMM cluster shows that disordered motifs are more evenly spread across all of the clusters. This comparison suggests that conventional heuristics are consistent with the actual distribution of structures in well-characterized proteins when it comes to well-defined sheet and helical motifs. On the other hand—at least when seen through the lens of the Ramachandran angles—DSSP bends, turns and coils are not clearly identifiable with separate peaks in the observed probability distribution. There are nevertheless clusters that are associated with clear peaks in the feature space that are not associated with helices or strands. This suggests that "disordered" sections of proteins exhibit substantial order on the scale of the conformation of individual residues, and that looking at the statistics and correlations of these local motifs might be a better approach to characterize disordered polypeptides than trying to fit them within existing categories.

One can further contextualize the probability distributions with the framework of the Q3 or Q8 score. Assigning Cluster 1 to the "strand" classification, Cluster 3 to the "helix" classification, and associating all other clusters with the "coil" designation (see Fig. 6.7) yields a Q3 score of 0.70 relative to DSSP and 0.72 relative to STRIDE. The rather low value of the Q3 score is comparable to the reported match scores of DISICL [182] (with our PAMM PMI-based method performing better relative to DSSP but more poorly relative to STRIDE), which is also based solely on backbone dihedral angles. However, the Q3 score of our cluster-based secondary structure assignments is substantially lower than other methods that rely on dihedral angles in addition to amino acid sequences [189, 198], or C_α distances [176]. In this context, the underperformance of our method in classifying secondary structure could be given two different justifications. One is that the traditional secondary structure motifs are based on rather arbitrary thresholds, which recognize configurations as separate modes even when there are no clearly distinct maxima in the distribution of atomic configurations, regardless of the (reasonable) choice of input representation. Another is that our specific choice of representation, i.e. pairs of backbone dihedrals, is insufficient to distinguish between different motifs because of its excessive locality. The latter hypothesis is supported by the large overlap of different DSSP motifs in dihedral space (Fig. 6.8), and can be tested by using different feature representations as the input to a PAMM analysis.

As a means of including more non-local information into the model while relying on a representation based purely on dihedrals, we also performed a PAMM clustering on the dihedral angles of consecutive residues, comparing the cluster assignment to the DSSP and STRIDE secondary structure classifications of the middle residue in the sequence. Just as in the two-dimensional case, in six dimensions (three consecutive residues) and ten dimensions (five consecutive residues) the helices and strands are localized to one or two clusters, while the other secondary structures are distributed across several clusters. (The probability distributions for the six- and ten-dimensional clusterings are given in Appendix B.) As a consequence,

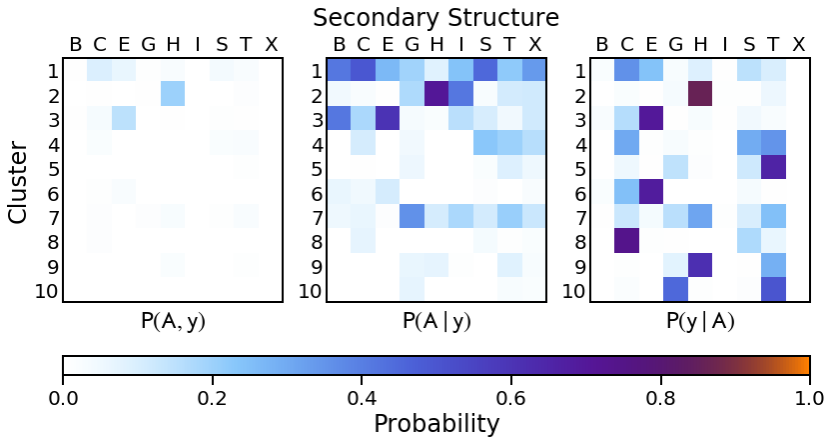

Figure 6.11 – Joint and conditional probabilities for the PAMM clustering of the first two principal components of the reduced SOAP vectors describing each residue of the protein backbone, where A is the PAMM cluster assignment and y is the DSSP secondary structure classification. Reproduced from Ref. [160] under CC BY 4.0.

the Q3 score is largely the same among the two-, six-, and ten-dimensional representations (see Table 6.3). Moreover, we observe that the Q3 score can be sensitive to the choice of clustering parameters; relatively small changes to the parameters can change the resulting GMM such that the Q3 score increases or decreases by $\approx 0.05 - 0.10$. For example, reducing the quick shift parameter from 0.90 to 0.80 in the ten-dimensional case roughly doubles the number of clusters and the Q3 score increases from approximately 0.68 to 0.73 for both DSSP and STRIDE.

To further explore the hypothesis of an insufficiently descriptive feature space serving as the reason for the discrepancy between the PAMM-based motifs and the conventional secondary structure definitions, we additionally analyzed the joint and conditional probability distributions for motifs in the SOAP feature space. Because each individual reduced SOAP vector is based on an expansion around the C_α atoms, each vector corresponds to a single residue and therefore can be associated with a DSSP- or STRIDE-assigned secondary structure, just as the dihedral angle representations, and the probability distributions and Q scores can be computed similarly. The joint and conditional probability distributions of the clustered SOAP vectors in the 2D feature space and DSSP secondary structure assignment are given in Fig. 6.11. (The probability distributions relative to the STRIDE assignments can be found in Appendix B, as well as those for the higher-dimensional SOAP-based representations.) Compared to the dihedral angle probability distributions, the distributions based on a clustering of the SOAP vectors are more diffuse. Instead of the helices and strands being confined to one or two clusters as with the dihedral angles, in the SOAP clustering the helices and strands are divided among several clusters. However, from the perspective of the Q3 score, the SOAP representation performs as well as the dihedral angle representations, with scores in the range of 0.70–0.74 for the two-, six- and ten-dimensional representations based on the principal components of the SOAP vectors. Fig. 6.11 shows that even with the more complete

description of local atomic environments provided by the SOAP-based description, there still exists a lack of correspondence between the motifs in the feature space and conventional secondary structure definitions. The fact that increasing the complexity of the environment descriptors does not improve the match between PAMM PMIs and conventional secondary structure motifs suggests that the discrepancy is not due to lack of descriptive power, but to the fact that conventional motifs are not reflected in the environment distributions observed in the PDB.

6.3.6 Supervised classification

To substantiate the hypothesis that conventionally defined structural motifs are not entirely representative of the structural space inhabited by the proteins of the PDB, we can train a *supervised* classification model to recognize DSSP or STRIDE motifs, as the performance of the classification model gives an indication as to whether or not our feature representations can adequately describe the structural features composing conventional secondary structure definitions. To perform this classification task we used a support vector machine (SVM) [128] as implemented in the `scikit-learn` Python package [202] to perform multiclass classification on the dihedral angle and SOAP-based representations, using as the classification targets the DSSP or STRIDE secondary structure labels. For each SVM model, we employed a "one vs. one" classification scheme [203] with regularization parameter $C = 1.0$ and a Gaussian kernel having width $\gamma = 1/N_f$, where N_f is the number of features. Furthermore, the SOAP PCA and dihedral angle data were scaled to have zero mean and columnwise unit variance before building the SVM. Of the approximately 4.3 million residues present in our dataset, we selected 200,000 residues at random (excluding those that were not assigned a secondary structure by DSSP or STRIDE) to train and evaluate the SVM. Of these 200,000 residues, 50,000 were randomly selected to serve as the training set, and the remaining 150,000 served as the test set. The Q3 and Q8 scores resulting from SVMs built on the reduced SOAP representation and the dihedral angle representation at various dimensionalities are given in Table 6.3, and are seen to improve systematically when the dimensionality of the representation is increased—contrary to what we observed with a PAMM analysis.

Fig. 6.12 show the learning curves of the Q3 and Q8 scores relative to DSSP for the multiclass SVM. Each point in each curve is an average score over five separate constructions of the SVM, each time using a new random subset of 200,000 residues. As learning saturates more quickly for the descriptors of lower dimensionality, the asymptotic (large train set size) classification accuracy of the supervised model indicates the limit that can be achieved with a given environment representation. Learning curves and tabulated Q3 and Q8 scores for SVM classification of STRIDE secondary structures is given in Appendix B.

The improving Q3 and Q8 scores for the dihedral angles and reduced SOAP representations in the SVM models coupled with the lack of obvious improvement in the cluster-based Q scores confirms that the limiting factor in the association between motifs is intrinsic to unsupervised learning. The reference heuristics—the DSSP and STRIDE secondary structure definitions—are simply not well-represented in the probability distribution of the data in the feature spaces that we use.

Table 6.3 – Q3 and Q8 scores relative to DSSP for PAMM PMI and SVM predictions of secondary structure based on a PCA of SOAP vectors and dihedral angles at various dimensionalities. The reported SVM scores are an average over five separate constructions of the SVM, each time using a new random subset of 200,000 residues, with 50,000 of these serving as the training set.

	PAMM PMI		SVM	
Representation	Q3	Q8	Q3	Q8
ϕ,ψ (2D)	0.71	0.61	0.78	0.67
ϕ,ψ (6D)	0.74	0.63	0.87	0.80
ϕ,ψ (10D)	0.73	0.61	0.88	0.82
SOAP PCA (2D)	0.73	0.58	0.75	0.61
SOAP PCA (6D)	0.72	0.58	0.84	0.73
SOAP PCA (10D)	0.71	0.55	0.90	0.79
SOAP PCA (100D)	—	—	0.95	0.89

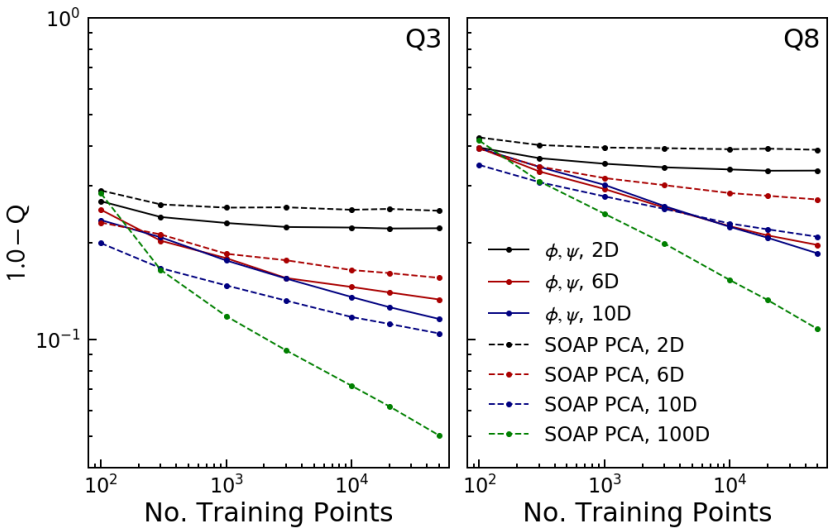

Figure 6.12 – Learning curves of Q3 and Q8 scores relative to DSSP for the multiclass SVM based on backbone dihedral angles and a PCA of the SOAP representation with various degrees of information content (i.e., the dimensonality of the descriptor). The Q scores are represented in the learning curves as errors, i.e., $1-Q$. Reproduced from Ref. [160] under CC BY 4.0.

This simple example highlights both the difference in unsupervised and supervised learning methods while also emphasizing the importance of the choice of feature representation. A supervised learning scheme is well-suited to adapt an existing motif definition to a different representation of atomic environments, and—in the limit of a sufficiently large train set—serves as proof of whether the chosen representation is sufficiently complete to achieve an accurate classification. An unsupervised clustering model, on the other hand, is useful for finding new patterns in feature space. Provided that the representation is complete, it also can serve as validation for established pattern recognition heuristics, showing whether the presence of well separate motifs is robust to the choice of structural representation.

6.4 Conclusions

The work presented in this chapter serves as a demonstration for how supervised and unsupervised learning can be used together to provide a more complete picture of the structural patterns present in materials. We began by showing that unsupervised learning techniques, namely Gaussian mixture models, are a flexible, generally applicable means of identifying motifs in materials. While conventional, heuristic-based motif definitions require additional domain-specific knowledge in order to capture the relevant motifs and avoid the specious, unsupervised machine learning approaches often "do the right thing" and automatically highlight statistically meaningful patterns with only minimal prior knowledge of the feature space.

We analyzed the differences between traditional and data-driven definitions of hydrogen bonds and secondary structure in experimental protein structures from the Protein Data Bank to show that there can sometimes be large discrepancies between the conventional and unsupervised motifs. In the case of the hydrogen bonds, we found the discrepancy was largely due to imprecision and the manual intervention required to discard specious motifs from the traditional distance–angle definition. For the case of protein secondary structure, we found notable mismatches between the modes in increasingly complex feature spaces and the patterns prescribed by the DSSP and STRIDE secondary structure assignments. Using support vector classification, we showed that, despite the lack of correspondence between the conventional and data-driven motifs, the conventional secondary structure labels could be predicted rather well using feature representations based on sequences of backbone dihedral angles and a PCA of the SOAP representation, suggesting that such structural descriptions capture the relevant features used to define conventional secondary structure motifs, and that the conventional notions of secondary structure in proteins do not map directly to modes in the structure space. The idea of using supervised learning to validate unsupervised representations is pursued further in Chapter 7, where both approaches are combined in more complex ways to appraise hypothetical zeolite frameworks for experimental synthesis.

7 Exploration of Zeolite Structures [1]

7.1 Introduction

In Chapter 6, we established how unsupervised learning could be used to identify statistically important structural motifs in a database of protein structures, using supervised learning to quantify how well the feature space is able to represent conventional motif definitions. In the present example, we demonstrate how supervised and unsupervised learning can be used to analyze classes of materials for which the prototypical motifs are more complex, varied, and difficult to represent with a general feature space. In particular, we evaluate several structural descriptors in their ability to serve as feature representations for machine learning predictions of the molar volumes and energies of zeolite frameworks. We use the resulting insight to construct a map of local atomic environments and to identify from a collection of hypothetical zeolites those that show the most promise for experimental synthesis.

Zeolites are nanoporous, crystalline silica-based materials primarily composed of corner-sharing SiO_4 tetrahedra, and may include heteroatoms such as Al, Ge, and P isomorphically substituted for Si sites. Because of their stable, porous frameworks and versatile structures and compositions, zeolites have found applications in gas storage [205, 206] and catalysis [207, 208]. Databases of real [209, 210] and hypothetical [54–56, 58, 59, 211] zeolites have previously been screened for applications-specific materials discovery efforts, such as carbon dioxide capture [212], but without any robust metric regarding the synthesizability of candidate structures, such screening exercises still leave much to be desired. To shed light on this, recent work has considered assembly [213, 214] of various rings [215] and cages [216] inspired by known or hypothetical zeolites. While such approaches are logical, enumerating zeolite substructures through the lens of known rings and cages can leave out many conceivable local silica environments not yet encountered, which may be synthesizable [211] and important for

[1] Sections 7.1 and 7.2, and their corresponding subsections, of this chapter are adapted with modifications from Helfrecht, B. A., Semino, R., Pireddu, G., Auerbach, S. M. & Ceriotti, M. A New Kind of Atlas of Zeolite Building Blocks. *The Journal of Chemical Physics* **151,** 154112. doi:10.1063/1.5119751 (2019), with the permission of AIP publishing; BAH performed the machine learning analyses and prepared the corresponding figures, RS and GP performed preliminary analyses and input processing and computed the classical descriptors, and all authors contributed to the design of the study and to the writing of the manuscript from which the present text has been adapted. Section 7.3 and its corresponding subsections contain work currently in preparation for submission; for this work, BAH performed the machine learning analyses and prepared figures, and all authors (Helfrecht, Semino, Pireddu, Auerbach & Ceriotti) contributed to the design of the study.

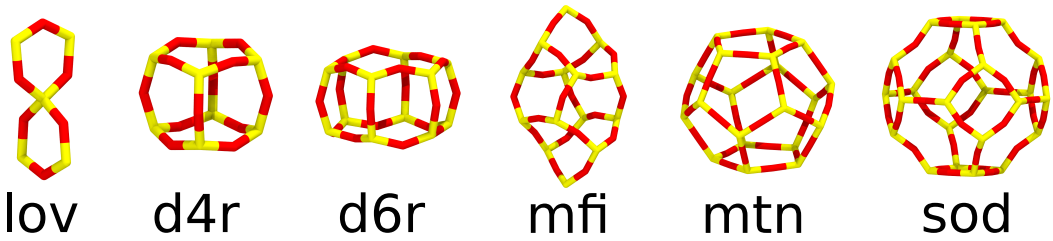

Figure 7.1 – Examples of composite building units.

investigating disordered silica structures leading up to zeolite crystals.

Zeolite structures are currently understood in terms of their framework topologies, which are idealized descriptions of the connectivity of the corner-sharing tetrahedra. These topologies are analyzed, in turn, by identifying rings that form each structure, where an "n-ring" involves n alternating –Si–O– atomic units. At the time of this writing, there are 242 topologies (of which 230 are fully connected) of fully ordered zeolite materials in the database of the International Zeolite Association (IZA) [210] and the accompanying Atlas of Zeolite Structure Types [209]. In the IZA database there are 36 topologies that can be synthesized as all-silica zeolites, i.e., that are polymorphs of α-quartz.

7.2 A Map of Zeolite Environments

At its core, the Atlas of Zeolite Structure Types is a *dictionary* of zeolites organized alphabetically by a set of arbitrarily assigned three-letter codes. Each entry includes information about the basic structural motifs present in the framework known as *composite building units* (CBUs), which often take the form of ring-like or cage-like objects centered on empty space. (Some examples of CBUs are given in Fig. 7.1.) The present organization of the Atlas can be very useful if one seeks structural information about a particular, known zeolite, but is less useful if one seeks to discover other zeolites with structural features similar to a certain zeolite. Furthermore, CBUs are often defined and identified by inspection and can be difficult to systematically represent in a numerical form for computational discovery efforts. To address these shortcomings, we develop a map of zeolite environments wherein nearby entries share similar structural features. To construct this map, we define descriptions of local, Si-centered environments in analogy with CBUs, using local geometric features in addition to the SOAP representation. We focus on atom-centered descriptions as they facilitate the mathematical reconstruction of zeolite frameworks and their properties through summations over atoms instead of void spaces.

7.2.1 Data Selection

Our analysis of zeolite structures and subsequent mapping of local environments is based on the Deem SLC PCOD database [56] (hereafter referred to as the "Deem database"), which contains hypothetical zeolite structures that are no more than 30 kJ/mol Si higher in energy than α-quartz as computed with the shell-model-based Sanders–Leslie–Catlow (SLC) forcefield [217, 218] in the program GULP [219].

Given that the Deem database contains a few hundred thousand hypothetical zeolites, each zeolite contains several Si-centered environments, and that a given feature representation can contain thousands of components and/or require significant computational resources, computing and analyzing the full SOAP vectors of every environment in the database is impractical, if not computationally intractable. Hence, we reduced the dimensionality of the input space by considering a subset of 10,000 structures from the approximately 330,000 structures in the database. The set of 10,000 structures was selected at a fixed stride according to ID number, which results in a diverse sampling of the spacegroups possessed by the frameworks. In addition, we considered a subset of 1,000 stride-selected structures from the Deem database to test whether our results are influenced by the size of our selected subset. We found that the results for the 1,000-structure sample yield similar conclusions to those that can be drawn for the 10,000-structure sample, indicating that our results are likely converged with respect to the structural diversity in the Deem database, and thus that our results can be generalized to the full database. In this chapter we focus on our findings based on the 10,000-structure subset; the results for the 1,000-structure subset can be found in Appendix C.

7.2.2 Environment Descriptors

In the context of zeolite structure analysis, the use of geometric descriptors for classifying and rationalizing structure–property relationships is already well-established [209, 220–222]. The choice of the representation for a given zeolite is often motivated by physical and chemical understanding and intuition regarding which structural features are relevant to the study of certain properties. For example, when investigating catalytic properties of zeolites, one may consider correlating acid-site strengths with Si – O – Al angular distributions [223]. On the other hand, when considering diffusion of guest molecules through zeolite frameworks, one typically uses zeolite ring distributions to rationalize transport properties [224].

As the first step in constructing our map of zeolite environments, we must decide on a numerical representation of the relevant structural features. Traditionally, the diversity of zeolites is often characterized by distributions over descriptors such as Si – O – Si angles [225], Si – Si near-neighbor distances, and ring sizes [54–56, 224]. We refer to these conventional representations as *classical descriptors* to emphasize the difference between these zeolite-specific descriptors and more generally applicable representations such as SOAP, against which we benchmark their performance. Schematic representations of the four atom-centered descriptors that we consider for analyzing local zeolite environments are given in Fig. 7.2, and the descriptors themselves are described in detail in the following subsections.

Classical Descriptors

Our set of classical descriptors is based on three widely used representations of chemical environments in zeolites, namely Si – O – Si angles, Si – Si distances, and connected rings. The distance- and angle-based descriptions are local, atom-centered features, and are appropriate for representing properties that can be safely decomposed into additive, local contributions. For instance, bonding interactions in interatomic potentials are often expressed in terms of bond angles and distances [226, 227]. The distance- and angle-based representations

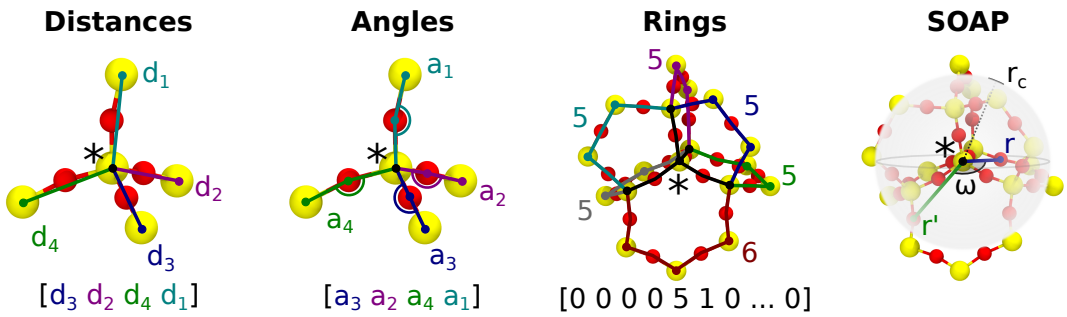

Figure 7.2 – Schematic depictions of the descriptors used to represent the hypothetical zeolite structures. A simple representation of the corresponding feature vector is given below each classical descriptor (those based on Si–Si distances, Si–O–Si angles, and ring counts). The SOAP feature vector is more complex, and can be understood as a three-body correlation function based on averaging over all rotations of a "template" consisting of two arms of length r and r' separated by an angle ω within a local atomic density with cutoff radius r_c.

characterize the local environment of each Si as a vector of the Si–Si distances or Si–O–Si angles between the central reference Si atom and the four nearest-neighbor Si atoms, as the structures we study are composed entirely of Si tetrahedra. In order to make these representations independent of permutations of the atom indices, the vector elements are arranged in descending order.

Ring-based descriptors have the potential to capture correlations on longer length scales, and consequently may be more suitable for characterizing the topology of a given framework than distance- or angle-based structural representations. Zeolite frameworks can be described in terms of rings according to various definitions; in order to be able to apply an automated analysis to a large database of structures, we base our descriptor on two mathematically rigorous definitions, namely King's criterion [222, 228] and the shortest path criterion [222, 229–231], as implemented in the R.I.N.G.S. code [222, 232]. In both cases, we translate the list of detected rings in a given zeolite framework into vectors of features x_s associated with local Si environments by counting the number of times the central Si atom appears in s-sized rings (since O atoms are also present in the rings, the atom-by-atom ring size is $2s$, but here we adopt the frequently used convention of naming a ring by counting only Si atoms). For example, the ring vector for a Si atom in silica-sodalite (which are all equivalent by symmetry) is $[0,0,0,2,0,4,0,\ldots,0]$, indicating that each Si atom is part of two 4-rings and four 6-rings. The ring descriptors that we compute are effectively ten-dimensional vectors, ranging from 3-rings to 12-rings: we find no rings smaller than size 3 and no rings larger than size 12. In the following we focus on the shortest-path definition, which gave marginally better performance than King's definition when used as the basis for predicting framework properties. A comparison of descriptors based on different ring definitions is given in Appendix C.

SOAP Descriptor

We additionally compute feature representations for local Si-centered zeolite environments using the SOAP power spectrum representation (see Section 2.1.2). We consider two different

representations, one employing a cutoff radius of 3.5 Å for the local environment, and another employing a cutoff radius of 6.0 Å, corresponding roughly to the first and second Si-neighbor distances, respectively. Both representations were computed with `quippy` [201], using 12 radial basis functions, a spherical harmonics band limit of 9, a cutoff transition width of 0.3, and an atomic Gaussian width of 0.3, which leads to SOAP vectors with approximately 3,000 elements. As noted in Section 7.2.1, such large descriptors can become unwieldy when used in conjunction with large collections of atomic environments. Consequently, to reduce the dimension of the feature representation, we selected the 500 most diverse SOAP vector components with FPS based on a random selection of 2,000 structures from the 10,000-structure subset. (The FPS components for the 1,000-structure subset were computed based on the full set of 1,000 structures.) The squared Euclidean distance between SOAP vectors was used as the distance metric for the FPS procedure [200]. The SOAP vectors were then computed for all 10,000 structures in the subset, but only the FPS components were retained. Each SOAP vector thus describes an atomic environment that comprises a central Si atom as well as all of the surrounding Si and O atoms within the cutoff radius. Oxygen atoms were not considered as environment centers. While we deal here with all-silica frameworks, the SOAP representation is equally applicable to zeolite frameworks with heteroatoms.

7.2.3 Machine Learning of Zeolite Properties

To objectively assess the performance of a given structural descriptor, we compute the performance of kernel-based regression models for predicting the molar volumes and energies of the zeolite frameworks using the descriptor as the model input. As the descriptors have different nominal size scales, we are also able to examine the degree of locality of each property, i.e., the correlation lengths that are required to determine the overall behavior of that particular property.

While machine learning models based on kernel methods can be quite powerful, they can also be computationally expensive, especially for very large datasets. When working with large amounts of data, as we do here, it can be useful to employ a low-rank approximation to the true kernel matrix built by considering the kernel between each environment and a set of representative environments. To this end, we used approximate kernel matrices constructed through the Nyström approximation to perform KRR and KPCA, applying the latter to further reduce the dimensonality of the SOAP-based descriptors for more level comparisons with the classical descriptors. Our low-rank kernel methods employ a Gaussian kernel and a set of 2,000 representative environments selected with FPS [106, 123, 200] [2].

In our regression models, we applied a scaling $\delta = M \times \mathrm{Var}(\mathbf{y}_N) / \mathrm{Tr}(\mathbf{K}_{MM})$ to the target property vector $\mathbf{y}_N$ and to each kernel matrix, and we included an additional regularization

[2] For some of the ring descriptors, the dataset contains fewer than 2,000 unique feature vectors, in which case only the unique representations were considered as representative environments. Additionally, to reduce memory requirements in determining the SOAP-based representative environments, the collection of structures in our dataset was divided into batches of 1,000, and 2,000 environments were selected by FPS from each batch. These selected environments were then concatenated into a single list from which 2,000 final environments were selected, again by FPS.

$\lambda_2 \mathbf{I}_{MM}$ so that the solution to our low-rank KRR model is,

$$\mathbf{w}_M = (\lambda_1^2 \delta \mathbf{K}_{MM} + \lambda_2 \mathbf{I}_{MM} + \delta^2 \mathbf{K}_{NM}^T \mathbf{K}_{NM})^{-1} \delta^2 \mathbf{K}_{NM}^T \mathbf{y}_N, \tag{7.1}$$

where λ_1 and λ_2 are regularization parameters, and $\mathbf{y}_N$ is a vector containing the known structural properties for the N structures. The kernel matrices $\mathbf{K}_{NM}$ are constructed according to Eqn. 2.19, where the entry corresponding to a particular structure is expressed as a sum over the kernel values associated with its constituent environments. The parameter λ_1 was optimized jointly with the Gaussian kernel width via five-fold cross validation on the full set of structures in the 1,000- or 10,000-structure subsets with the aim of minimizing the mean absolute error (MAE) of the regression; λ_2 was set to $10^{-16} \times \sigma$, where σ is the largest eigenvalue of $\lambda_1^2 \delta \mathbf{K}_{MM} + \delta^2 \mathbf{K}_{NM}^T \mathbf{K}_{NM}$. The same fold split was used across all models. The Gaussian kernel width for the KPCA decompositions of the SOAP-based descriptors was set to the optimized width determined for the predictions of the molar volume. We further constructed regression models for predicting the molar volumes and energies using as input various numbers of KPCA components of the SOAP descriptors. For these models, hyperparameters were optimized using 500 principal components (the same dimensionality as the full SOAP vector).

In our KRR models for which the target property is the framework energy, we additionally subtracted the mean energy per atom across all frameworks from the total framework energies, and learn the centered energies on a per Si basis. Conversely, the framework volumes were not centered before serving as inputs to our machine learning models for predicting the volume per Si atom.

Framework Properties

To visualize the performance of the SOAP-based and classical descriptors in representing zeolite structure–property relationships, we present a series of learning curves showing the MAEs for the property estimation exercises as functions of the number of training points. Building learning curves for the prediction of molar volume and energy enables head-to-head comparisons of the information content in the various structural descriptors and offers insights into the completeness and directness of a representation for describing a particular property or set of properties [8, 12, 78]. In particular, the value of the MAE for small training set sizes indicates whether the most prominent components of a representation correlate strongly with a given property. The asymptotic behavior for large training set sizes indicates how complete a representation is: the saturation (flattening to zero slope) of a learning curve indicates that the learning potential of the feature representation has been exhausted, and that predictions will cease to improve as additional training samples are added; conversely, a substantial slope indicates that the model still has sufficient information to improve its learning as the size of the train set is increased.

We report learning curves based on five-fold cross validation, as our main goal is to make predictions specifically for the hypothetical zeolites in the Deem dataset rather than generalizing to all possible zeolite structures, in which case an independent test set may be used to compute the learning curves. The learning curves employed the same fold splits as the

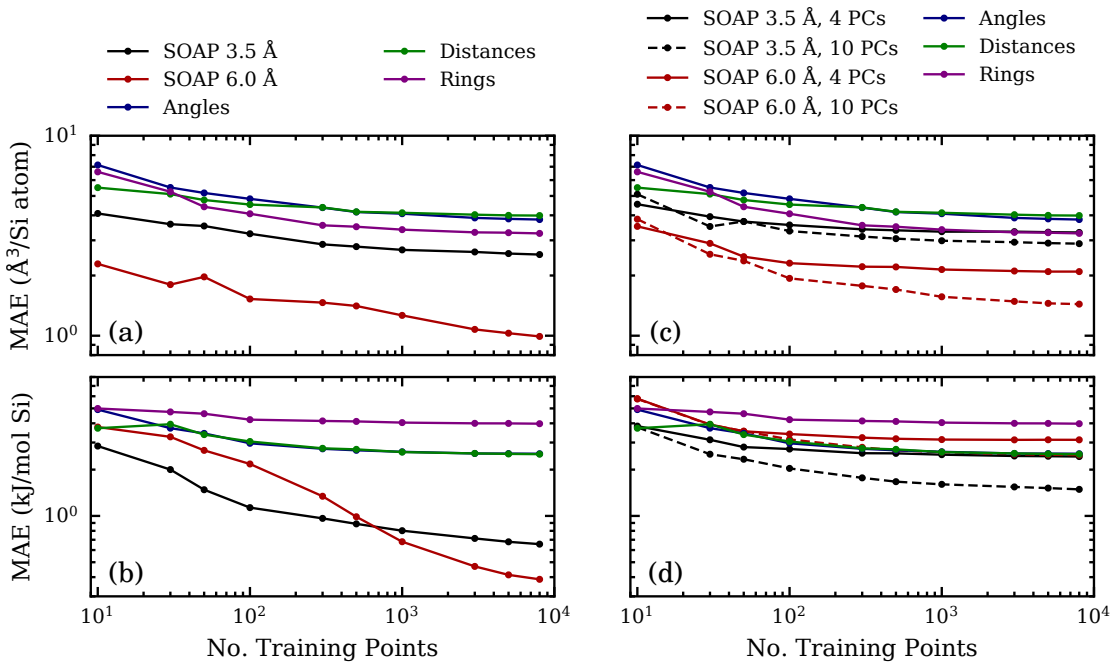

Figure 7.3 – Learning curves of the classical and SOAP descriptors for predictions of (a) volume per Si atom and (b) energy per mol Si. The error for each point in the learning curve calculated as the average of a five-fold cross-validation procedure using the optimal regularization and Gaussian kernel width. (c) and (d) re-plot the learning curves of the classical descriptors alongside the SOAP-KPCA descriptors with similar dimensionality (i.e., the number of features composing the representation). Adapted from Ref. [204] with permission of AIP publishing.

hyperparameter optimizations; at each iteration of the cross validation for each training set size n_{train}, the model was trained on n_{train} samples randomly selected from the corresponding train fold and the property MAEs were evaluated on the corresponding test fold.

Figures 7.3(a) and 7.3(b) show the learning curves for predictions of the molar volume and molar energy, respectively, using the full classical and SOAP descriptors as described in Section 7.2.2. All three classical descriptors, based on Si–Si distances, Si–O–Si angles, and shortest-path ring counts, perform poorly at predicting both volume and energy, saturating at MAEs above 3 Å^3/Si atom and 2 kJ/mol Si, respectively. The ring-based descriptor is marginally better at predicting volumes compared to the distance- and angle-based descriptors, but it is the worst classical descriptor for predicting energies. This is likely because the ring-based descriptor only describes the distribution of ring sizes in a framework and does not account for distortions or other geometrical factors to which the lattice energy is sensitive.

Using the SOAP descriptor with a 6.0 Å cutoff results in the best volume predictions for all training set sizes, with MAEs as low as 1 Å^3/Si atom. This is probably due to the fact that accurate predictions of overall framework densities require information on larger spatial scales. The SOAP descriptor with a 3.5 Å cutoff performs only slightly better than the classical descriptors in predicting volumes, but yields the best energy predictions among all the descriptors for smaller training set sizes, indicating that relatively local correlations are

sufficient for making estimates of lattice energy to within approximately 1 kJ/mol Si. This is perhaps not surprising, since a substantial contribution to the zeolite lattice energy can be accounted for through nearest-neighbor bonding geometries, and thus the 3.5 Å based model, where only the first-neighbor tetrahedral information is included, yields reasonably accurate energy estimates even when the training set has a modest size. However, the learning potential of the 3.5 Å SOAP descriptor plateaus as the training set size approaches 1,000 structures. The 6.0 Å SOAP representation continues to improve at larger training set sizes, yielding energy predictions with MAEs as low as 0.4 kJ/mol Si, even though it begins to saturate towards 8,000 training frameworks, as it cannot fully describe long-range electrostatic interactions; in order to account for such interactions, a nonlocal feature representation such as LODE [233, 234] may be required. Overall, Figures 7.3(a) and 7.3(b) show that the SOAP descriptor with a 6.0 Å cutoff does an excellent job of accurately capturing zeolite energy and molar volume, while the classical descriptors do not.

To investigate whether the improved predictive performance and learning ability of the SOAP descriptor is a result of the intrinsic quality of the descriptor itself, or merely of the fact that the SOAP representation incorporates more information through higher-dimensional vectors, we compare the learning curves of the classical descriptors with those of the SOAP descriptors whose dimensionality, i.e., the number of features composing the representation, has been reduced through KPCA. Because the distance- and angle-based descriptors are represented as four-element vectors (because of the tetrahedral coordination around Si) and the ring vectors contain ten distinguishing elements (because the ring sizes in our data set range between 3 and 12), we find it instructive to compare the classical descriptors with the first four and ten principal components of the SOAP representations, shown in Figs. 7.3(c) and 7.3(d). For predictions of the molar volume (Fig. 7.3(c)), the SOAP-KPCA descriptors tend to perform better than the classical descriptors regardless of the dimensionality of the representation, though the performance gain of the 3.5 Å SOAP-KPCA descriptors over the classical descriptors is rather small or even nonexistent at large training set sizes. For predictions of the molar energy (Fig. 7.3(d)), the ten-dimensional 3.5 Å SOAP-KPCA descriptor performs the best, and the four-dimensional 3.5 Å SOAP-KPCA descriptor performs quite similarly to the classical descriptors. The 6.0 Å SOAP-KPCA descriptors, on the other hand, perform worse than the distance- and angle-based descriptors at comparable dimensionality, but better than the ring-based descriptor. The behavior of 6.0 Å SOAP-KPCA can be attributed to the fact that 6.0 Å SOAP considers a larger local environment, and therefore, a much larger amount of information about the surroundings of a single Si atom. Because KPCA is not guaranteed to select the correlations that are most relevant to the regression [151], achieving accurate predictions requires the inclusion of a larger number of components in order to ensure that the most important structure–property correlations are accounted for. Indeed, in predictions of the molar energy, we find that the 3.5 Å SOAP representation outperforms the 6.0 Å SOAP representation for a given number of principal components unless upwards of 300 components are included. (Learning curves for the SOAP-KPCA representations with a larger range of principal components are provided in Appendix C.) Overall, we note that the SOAP-KPCA models generally equal or surpass the performance of the classical descriptors

at predicting the molar energies and volumes, even after dimensionality reduction. This suggests that SOAP inherently contains more information about the local structure of a zeolite framework than do the classical descriptors, and the more accurate property predictions are not merely a result of the flexibility afforded by the higher dimensionality of the feature representation.

Environment Property Contributions

Using the additive relationships for kernels between structures and environments outlined in Section 2.4, the optimal weights $\mathbf{w}_M$ (Eqn. 7.1) from our low-rank KRR models can be used to "decompose" the known structural property values $\mathbf{y}_N$ into contributions from the n individual atomic environments $\hat{\mathbf{y}}_n$ across the whole dataset [108],

$$\hat{\mathbf{y}}_n = \mathbf{K}_{nM}\mathbf{w}_M. \tag{7.2}$$

This allows us to use the machine learning models described in Section 7.2.3 to examine structure–property relationships on the scale of local atomic environments based on the known properties of whole structures. For the purpose of this exercise, the KRR models are trained on the full subset of 1,000 or 10,000 structures, and the property contributions are computed for all environments in the subset.

In Fig. 7.4 we examine the relationship between the reference framework properties and the decomposed environment properties. The middle panel of Fig. 7.4 shows a KDE in volume–energy property space for the Deem frameworks and their constituent environments, representing the probability that a particular framework or environment possesses a particular combination of molar volume and energy. The distribution of environments in the volume–energy property space is unimodal and reflects a continuum of property combinations. While the distribution of environment energies is much broader than that of the frameworks, both appear to share the low-energy "edge" highlighted by the thick red line to guide the eye. This energy–density correlation has already been noted by Pophale et al. [56] in their description of the Deem database, where they also note that the framework properties of real, known zeolites tend to lie along this "edge" in the property space [235], which has been attributed to the limitations of current solution-based synthesis approaches [236].

Given the broader distribution of environment properties compared to those of the whole frameworks, it is instructive to examine how the environment properties manifest themselves within a given framework. The atomic snapshots in the top and bottom panels of Fig. 7.4 provide two notably different examples: the top panel shows the atomic structure of the framework containing the median-energy environment in our 10,000-structure subset (framework A), and the bottom panel shows the framework containing the highest-volume environment (framework B). Framework A has a rather homogeneous structure, and as a consequence the individual environments possess a smaller range of property values. In contrast, framework B comprises environments with a wide range of property values that vary smoothly throughout the atomic structure. There also exists a clear, qualitative correlation between the local, Si-centered volumes and environments within framework B that is not as

easily discernible in framework A as a result of its higher relative homogeneity. Finally, we note that the highest-volume environment of framework A borders a very large pore, and that the local volumes surrounding the pore decrease as one moves towards the ends of the pore that have smaller radii of curvature and where the local atomic density is greater, serving as an intuitive check on our decomposition of structural properties into local contributions.

The qualitative correlation between the environment properties and local structural characteristics evident in Fig. 7.4 raises the question of whether such correlations can be quantified. To answer this question, we take advantage of the the results of Fig. 7.3, which shows that the SOAP descriptor is able to effectively account for the molar volume and energy of zeolite structures, even when truncated through KPCA.

In Figs. 7.5(a) and 7.5(b), we show the the relative variances σ^2_{KPCA} of the kernel principal components (KPCs) and their Pearson correlation coefficients with the environment energies $\rho_{KPCA,E}$ and volumes $\rho_{KPCA,V}$ for the first 50 components of the 3.5 Å and 6.0 Å SOAP descriptors, respectively. By construction, the KPCs are sorted in decreasing order according to the level of variance in the original data that they can individually account for. For the 3.5 Å descriptor, the first three components, highlighted with open circles, account for 52%, 13%, and 6% of the total variance, together explaining 71% of the structural diversity encoded in the SOAP vectors as measured by the component-wise variance. For the 3.5 Å descriptor, the first component is the most descriptive by far, with the second component having a relative variance of only 0.25.

In contrast, all three of the top components for the 6.0 Å descriptor capture a substantial portion of the structural diversity, with relative variances all exceeding 0.6. More specifically, the first component of the 6.0 Å SOAP descriptor explains approximately 24% of the variance; the second and third components each account for an additional 18% and 16% of the variance, respectively, so that 58% of the diversity is accounted for by the first three components. The lower explained variance ratio for the first three components of the 6.0 Å descriptor compared to the 3.5 Å descriptor can again be attributed to the fact that the "information content" of the 6.0 Å descriptor is larger and cannot be condensed as efficiently.

We can additionally examine the Pearson correlation coefficients between the KPCs and the environment properties to quantify the correlations between the structural features encoded in the SOAP vectors and the local volume and energy contributions. The Pearson correlation coefficients for the 3.5 Å SOAP representation in Fig. 7.5(a) indicate that the first component correlates reasonably well with both the energy and volume, but the next few components show somewhat weaker correlations. The fourth and seventh components then correlate rather well with the environment volumes, and the ninth component with the environment energies, emphasizing the fact that even components with low relative variances can be important in regression tasks [151]. Compared to the 3.5 Å SOAP representation, the low-index components of the 6.0 Å descriptor more consistently exhibit stronger correlations with the environment properties. Fig. 7.5(b) shows that KPCs 1 and 2 correlate strongly with volume, while KPCs 2 and 3 correlate strongly with energy. Overall, Fig. 7.5 suggests that a three-dimensional picture of structural diversity using the first three KPCs can reveal the essential features of the Deem zeolite data set.

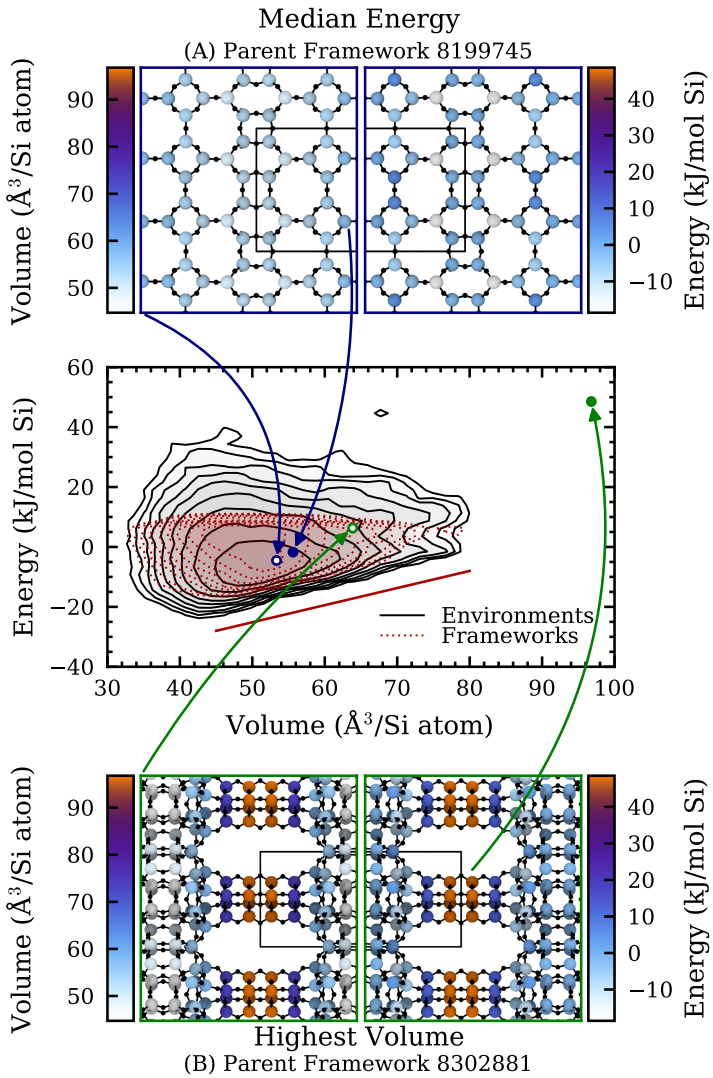

Figure 7.4 – Kernel density estimation of all environments in the 10,000-structure sample in energy–volume space (middle). Atomic snapshots of the frameworks containing the median-energy (top) and highest-volume (bottom) environments are also provided; the locations of these environments and their parent frameworks in the volume–energy property space are denoted with closed and open circles, respectively. In the left half of each atomic snapshot, the Si atoms are colored according to their volume contributions to the parent framework; in the right half, each Si atom is colored by its energy contribution. Reproduced from Ref. [204] with permission of AIP publishing.

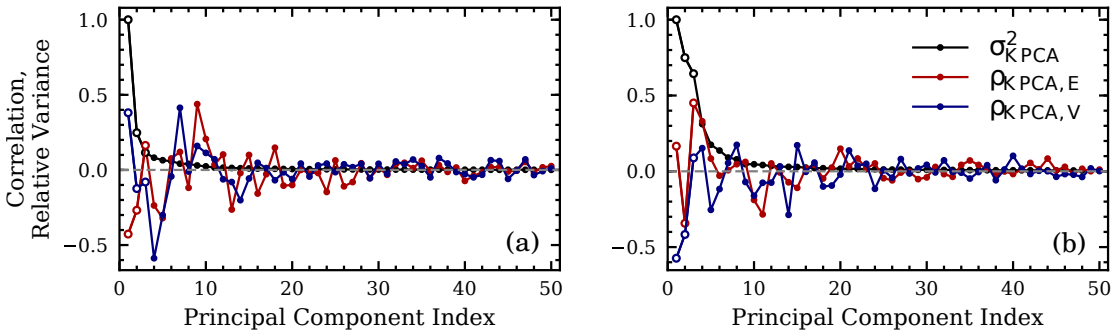

Figure 7.5 – Pearson correlation coefficients between the first 50 KPCs of the (a) 3.5 Å SOAP representation and (b) 6.0 Å SOAP representation and the decomposed environment volumes and energies in the 10,000-structure sample. The relative variance in the KPCs at each of the first 50 components is also plotted. The correlation coefficients and relative variance of the first three components are highlighted with open symbols. Adapted from Ref. [204] with permission of AIP publishing.

7.2.4 Mapping Zeolite Environments

Through the results presented in Section 7.2.3, we have shown that a SOAP-based representation can be used to build models for accurate predictions of the molar volumes and energies of hypothetical zeolites and that the same representation can be used to decompose structure-wide properties into contributions from individual atom-centered environments that both qualitatively and quantitatively correlate with local structural features. Consequently, we can use the KPCA decomposition of the SOAP feature vectors to construct an intuitive map of zeolite building blocks, analogous to the enumeration of CBUs in the IZA database [209, 210], and to the list of "natural building units" given by Blatov et al. [216]. The primary distinguishing feature of our approach is that we formulate our zeolite building blocks in terms of atom-centered representations, allowing such environments to be averaged to yield macroscopic properties of overall frameworks such as molar energy and volume. In principle, any set of atom-centered properties could be superimposed onto the map, though the existence of correlations between the properties and the KPCA-based coordinate system is not guaranteed; a map in which the projections correlate strongly with a set of reference properties can instead be obtained, by construction, through PCovR (see Section 5.1).

Our SOAP-based map of zeolite environments is presented in Fig. 7.6, where the environments are plotted in the space defined by the first three KPCs of the 6.0 Å SOAP feature vectors [3]. As a consequence of using the KPCs as the coordinate system of the map, the environments are naturally organized so that the distance between them is related to their structural similarity, and by extension (though to a lesser extent), their properties. The latter attribute stems from the fact that the first few KPCs of the 6.0 Å SOAP representation correlate with the local volumes and energies. In this way, our mapping scheme provides the possibility of finding similar zeolite environments by identifying a point of interest and searching the

[3] An interactive version of the environment map for the 1,000-structure set is provided as an example system for the interactive viewer `chemiscope` [237], available at https://chemiscope.org

surrounding area.

Each point (environment) in Fig. 7.6 is colored according to its energy contribution and is sized according to its volume contribution to its parent framework, and several environments are highlighted to provide examples of the "building blocks" present in the dataset. In particular, we show the atomic structures of the lowest-, median-, and highest-energy environments as well as the lowest-, median-, and highest-volume environments. In each highlighted structure, the atoms included within the 6.0 Å SOAP cutoff are shown in yellow (Si) and red (O), while the rest of the zeolite framework is depicted as corner-sharing tetrahedra. Contour plots showing the distribution of environments are projected onto the xy-, yz-, and xz-planes. These contour plots reveal that the statistical distribution of zeolite environments is unimodal and rather broadly peaked, indicating a remarkably uniform distribution of structural motifs. As such, we find no special region in the 3D KPC-space that is particularly well-stocked with building blocks for making hypothetical zeolites; our collection of environments thus represents a "continuum" of atomic substructures rather than discrete motifs. The broad distribution of environments shown in Fig. 7.6 further suggests that the algorithm for producing these hypothetical zeolite frameworks [54, 56] has left no substantial gap in environment space.

7.3 Candidates for Experimental Synthesis

The principal utility of the map in Fig. 7.6 is that the structural features plotted therein are organized by their structural similarity, facilitating comparisons between the structures and substructures present within the map. We can use this same idea in an effort to understand the similarities and differences between the hypothetical frameworks in the Deem database and the experimentally synthesized frameworks present in the IZA database and ultimately identify hypothetical frameworks that may be synthesizable. Operating under the assumption that the potentially synthesizable structures in the Deem database will exhibit structural similarities with those frameworks in the IZA database, we must first determine whether the structural space covered by the Deem frameworks overlaps with or wholly contains the structural space defined by the IZA frameworks. If the two spaces are distinctly different, we cannot hope to identify potentially synthesizable Deem frameworks through structural comparisons with the IZA frameworks. If the IZA and Deem structure spaces *do* overlap, we can then make an attempt to identify potentially synthesizable Deem frameworks based on our knowledge that the frameworks contained in the IZA database have indeed been successfully synthesized. To accomplish this task, we use a sequential workflow combining supervised and unsupervised learning to identify a relatively small number of frameworks from the Deem database that show the most promise as candidates for experimental synthesis. In particular, we use support vector classification to distinguish the Deem frameworks from the IZA, and subsequently construct a latent space based on a confidence measure of the classification. By constructing a convex hull in this latent space, we are able to identify those hypothetical frameworks that might be synthesizable.

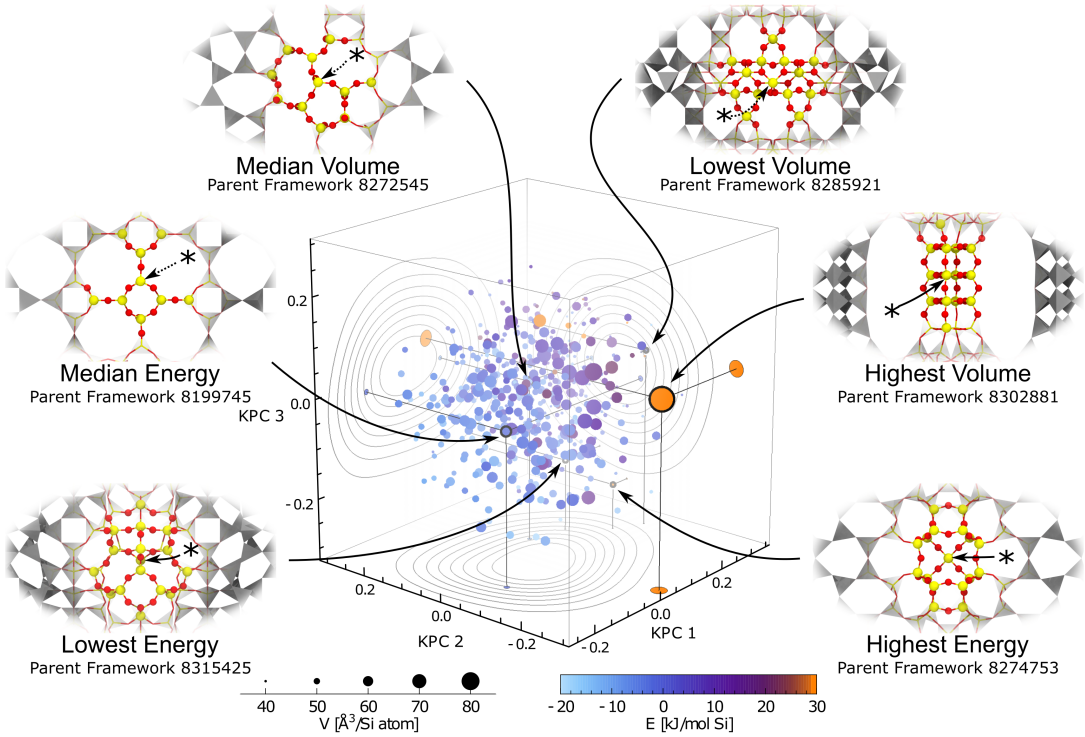

Figure 7.6 – A mapping of zeolite building blocks, where every 2,000-th environment of the 10,000-framework subset is plotted as a point in the three-dimensional space formed by the first three kernel principal components of the SOAP representation using a 6.0 Å cutoff. The points are colored and sized according to the energy and volume contribution of the corresponding environment to its parent framework. The environments with the highest and lowest energies and volumes are highlighted along with environments contributing energies and volumes close to the median of the dataset. Note that there exist some (extreme) outliers: the highest-energy environment contributes more than 380 kJ/mol Si, and the lowest below –30 kJ/mol Si. The highest-volume environment contributes more than 90 $Å^3$/Si atom, and the lowest less than 30 $Å^3$/Si atom. Energies falling outside the range of the scalebar are assigned to the color at the nearest extreme of the colorscale. Environment centers are indicated by the asterisks and their associated arrows; a dotted arrow signifies that the central atom is hidden behind the foremost atom visible in the atomic snapshot. In each snapshot, the atomic environment is represented as a ball-and-stick model; the surrounding zeolite structure is represented as SiO_2 tetrahedra. Overall, we see a remarkably uniform distribution of environments. Reproduced from Ref. [204] with permission of AIP publishing.

7.3.1 Data Selection

We search for potentially synthesizable frameworks in the entire Deem SLC PCOD database [56], which contains 331,172 frameworks in total. As our reference known synthesized frameworks, we use the all-silica analogues of the 230 fully connected frameworks in the IZA database. We further group the IZA frameworks into four "cantons" according to their reported reference composition as: (1) containing Si and O only; (2) containing Si and O, with potential substitutions; (3) containing O but no Si, e.g., aluminum phosphates; (4) containing neither Si nor O. Since not every framework in the IZA database has been synthesized with an all-silica composition, we relax all of the IZA frameworks with GULP [219] using the SLC forcefield [217, 218]. To ensure that the IZA and Deem frameworks are comparable in structure and energy, we attempt to replicate the relaxation procedure used for the Deem frameworks in Ref. [56]: we first attempt to optimize the IZA unit cell and atomic positions (both core and shell) under constant pressure conditions, and if this optimization does not converge, we perform from scratch a constant volume optimization. We additionally attempt to reproduce the lattice energies for the Deem frameworks as reported in the Deem database by optimizing only the shell geometries; we generally find success in this endeavor, with the MAE between the database energies and the energies obtained from our optimizations less than 0.1 kJ/mol Si. However, for five frameworks we find energy discrepancies of more than 10 kJ/mol Si, and we consequently discard these frameworks from our analysis as we have no guarantee that their structures and energies are compatible. A histogram of the energy errors is provided in Appendix C.

As the SOAP descriptor proved to be successful serving as a feature representation for the prediction of the molar volumes and energies of the Deem frameworks (Section 7.2.3), we compute the SOAP representation for all of the IZA and Deem frameworks as in Section 7.2.2, with a few important differences. The first is that we now use the `librascal` [93] package to compute the representation, which allows us to compute a high-quality radial basis through a spline approximation, as outlined in Section 2.1.2. This high-quality basis is constructed through a PCA decomposition of the density coefficients corresponding to 32 Legendre polynomials in the discrete variable representation, and we retain the top 8 components. As before, we use nine angular functions as well as a cutoff transition width of 0.3 and an atomic Gaussian width of 0.3. In contrast to the SOAP-based representation used to construct the map of local zeolite environments, here we do not subselect components from the SOAP feature vectors but rather use the full representation. The retention of all features and the use of a high-quality radial basis allows us to examine in greater detail the specific structural features that distinguish the IZA and Deem frameworks. For this same reason we focus in the following on linear models in contrast to the kernel methods employed in Section 7.2, allowing us to transparently examine the connection between those features that a particular model implicitly marks as important and the structural characteristics to which those features correspond. Our use of linear models thus allows us to define the SOAP features for an entire structure as an average over its constituent environments, as described in Section 2.4. In all of our models the training data are columnwise centered and are scaled to have a Frobenius norm of $\sqrt{n_{\text{train}}}$, with the test set centered and scaled relative to the train

set. Where we use cross-validation to optimize hyperparameters, this preprocessing is done independently in each iteration.

7.3.2 Comparison of Structure Space

As a first point of comparison between the hypothetical (Deem) and known synthesizable (IZA) frameworks, we compute a PCA of the 6.0 Å SOAP features for the zeolite structures, similar to the KPCA decomposition used to construct the map in Fig. 7.6, where instead of examining local environments we use representations of whole zeolite frameworks. The PCA is trained only on the subset of 10,000 Deem frameworks described in 7.2.1 so that the resulting IZA projections locate the known synthesized frameworks relative to the structural space defined by the Deem frameworks. Figs. 7.7(a)–(c) thus show histograms of the first three PCA component values for the IZA and Deem frameworks, clearly indicating that the structural space covered by the Deem frameworks encompasses that of the IZA frameworks, with the IZA frameworks lying at the edge of the Deem structural space. This suggests that our search for frameworks in the Deem database that share structural similarities with IZA frameworks could be confined to a small portion of the structural space—the majority of the Deem frameworks can immediately be discarded as unlikely to be synthesizable. However, just as we showed in Chapter 6 that machine learning methods prove robust to the retention of specious hydrogen bond configurations, we demonstrate in the following that we need not discard the obviously hypothetical frameworks in order to find success in identifying synthesis candidates. At the same time, due to the relative size of the Deem database, a large number of Deem structures lie within the IZA structural envelope, making it difficult to pare down the Deem database to a manageable number of potentially synthesizable structures based on a purely unsupervised PCA mapping. We address both of these issues in Section 7.3.3, where we use supervised classification techniques to more robustly compare the similarities and differences between the IZA and Deem frameworks. Figs. 7.7(d) and (e) show histograms of the molar volumes and energies of the IZA and Deem frameworks. Chemical intuition would indicate that the potentially synthesizable Deem frameworks are those that are lowest in energy and have molar volumes similar to those of IZA frameworks; however, such heuristics are blind to "unrealistic" structures that have, for example, very low energy but highly distorted tetrahedra.

We can further quantify the coincidence of the IZA and Deem structural spaces by comparing the predictions of molar volume and energy for the IZA and Deem frameworks from a ridge regression model again trained only on the subset of 10,000 Deem structures. The regularization of the regression model was determined through a grid search using five-fold cross validation to minimize the MAE on the validation set. The MAEs on the test set (the 230 IZA frameworks and 250 randomly selected Deem frameworks not in the train set) are provided in Table 7.1. Generally speaking, the predictions for the IZA frameworks are comparable to those of Deem under the 6.0 Å SOAP descriptor, but are substantially worse (though not unreasonable) for the 3.5 Å descriptor. Of particular note is the cantonal error breakdown: the observed errors tend to increase from Canton 1 (only Si and O) to Canton 2 (Si, O, and other species) to Canton 3 (O, no Si) as the frameworks become more compositionally dissimilar to the Deem structures. Canton 4 contains only a single structure (RWY), the composition of

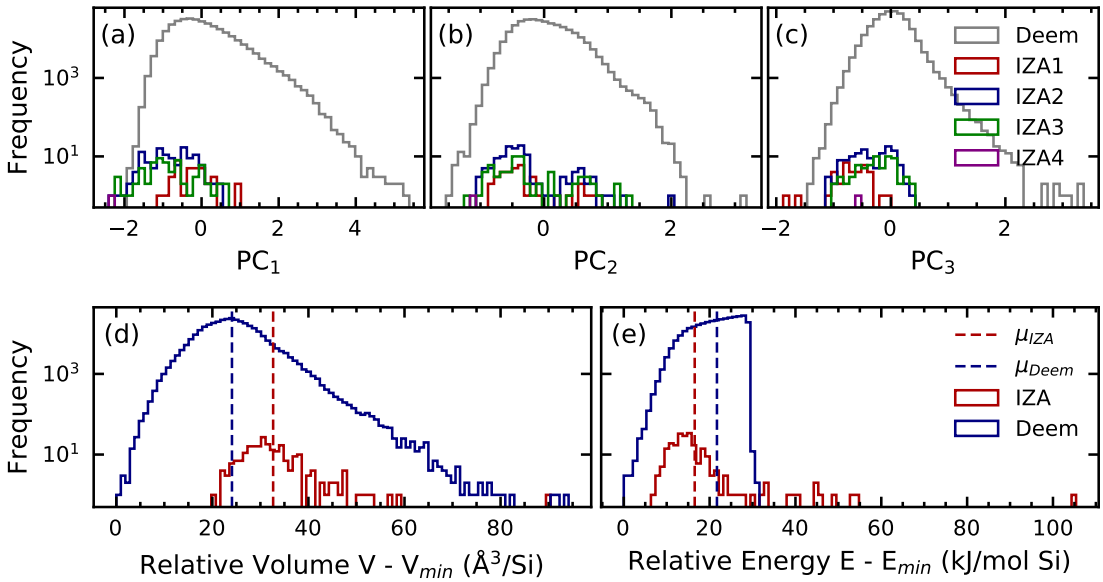

Figure 7.7 – Histogram of values of the first three principal components of the power spectrum SOAP vectors of a subset of 10,000 Deem frameworks and all 230 IZA frameworks. The histogram makes evident that the IZA frameworks are concentrated near the edge of the structural space defined by the Deem frameworks. The PCA projection is defined only by the 10,000 Deem frameworks.

which contains neither Si nor O, and unsurprisingly makes for difficult predictions. Part of this difficulty can be attributed to the fact that we compute the energy for RWY based on an all-silica analogue of its experimental structure with a forcefield tuned for frameworks that contain only Si and O. Indeed, we find that RWY has a much higher energy as computed by GULP than any of the other IZA structures, and for this reason we omit it from our subsequent analyses. A histogram of IZA energies as computed by GULP is given in Appendix C.

7.3.3 Synthesis Assessment Workflow

Having shown once again that the SOAP descriptor is capable of capturing the relevant structural features for making comparisons between the hypothetical Deem frameworks and the known synthesizable IZA frameworks, while also demonstrating a need for a more sophisticated approach for assessing the synthesizability of the Deem frameworks, we develop here a sequential workflow combining various machine learning methods to better understand the similarities and differences between the structures in the IZA and Deem databases with the ultimate aim of finding the synthesizable needles within the Deem haystack. A schematic of this workflow is shown in Fig. 7.8, where we begin by computing the SOAP feature representations and GULP energies of the frameworks.

Analysis of Classification Models

After computation of the SOAP vectors and energies, the entry point to this workflow is support vector classification, where we attempt to distinguish the IZA frameworks from the

Table 7.1 – Mean absolute errors (MAEs) for predictions of molar volume V (units Å^3/Si) and molar energy E (units kJ/mol Si) from a linear ridge regression model trained on a subset of 10,000 structures from the Deem database and tested on an unseen set of Deem and IZA structures. While IZA prediction errors can be 1.5–3× larger than for the Deem structures, the volume and energy predictions are not unreasonable, particularly for the all-silica structures and for the models based on the 6.0 Å SOAP representation.

		3.5 Å		6.0 Å	
	n_{test}	V	E	V	E
Deem	250	2.81	0.65	1.10	0.19
IZA	230	5.30	0.92	1.70	0.18
IZA1	36	4.54	0.98	0.96	0.14
IZA2	125	5.17	0.88	1.57	0.15
IZA3	68	5.38	0.94	1.91	0.23
IZA4	1	44.28	2.02	30.52	1.97

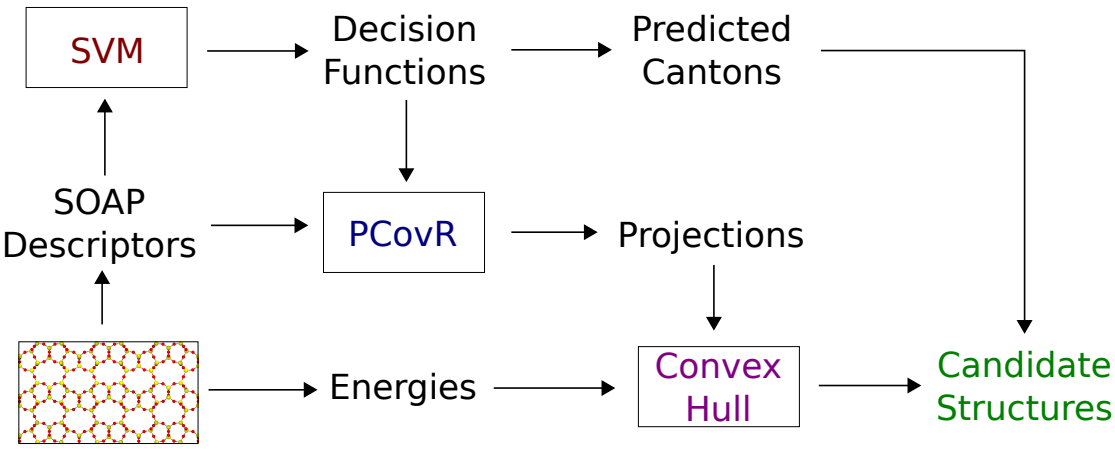

Figure 7.8 – Schematic of the SVM-PCovR-CH infrastructure. The GULP energies and SOAP descriptors are computed for each framework, and the SOAP descriptors are used as input to both SVM and PCovR models. The decision functions resulting from the SVM classification are additionally used as input to the PCovR model, where they are combined with the SOAP features to develop a latent space projection that serves as the basis for a convex hull (CH) construction using the GULP energies as a measure of thermodynamic stability. The structures near the convex hull can then be compared against the SVM classification predictions and corresponding decision functions to create a hierarchy of synthesis candidates.

Deem based solely on their SOAP feature vectors. To understand how different structural features impact the classification, we construct an ensemble of SVM models, each based on a different SOAP representation or subset of SOAP features corresponding to different atomic correlations. In particular, we examine both 3.5 Å and 6.0 Å SOAP representations to understand the spatial scale of the features most relevant for the classification as well as representations including only two-body correlations (the radial spectrum) or additionally three-body correlations (the power spectrum). For the radial spectrum representations, we build classification models based on only Si–O correlations, only Si–Si correlations, and both Si–O and Si–Si correlations. For the power spectrum representations, we examine separately Si–O–O correlations, Si–O–Si correlations, Si–Si–Si correlations, and all possible combinations thereof. For brevity, we label these models by omitting mention of the central Si atom common to all studied correlations and by using a "+" to denote a combination of correlations. For example, we use "OO+OSi" to label a model trained jointly on Si–O–O correlations and Si–O–Si correlations, and we use "Si" to label a model trained only on Si–Si correlations. We additionally consider both the binary ("IZA vs. Deem") and multi-class cantonal ("IZA1 vs. IZA2 vs. IZA3 vs. Deem") classifications, where in the multi-class case we employ a "one vs. rest" classification scheme.

For each classification task, the SVM is trained jointly on one half of the IZA frameworks (excluding RWY) and the 10,000-structure subset of Deem frameworks. Given that the Deem database contains a number of structures identical to those in IZA, we remove from the analysis those Deem structures we determine to be identical to IZA frameworks to avoid pairs of contradictory training labels. To find the Deem frameworks that are identical to IZA, we simply compute the Euclidean distance between their full 6.0 Å SOAP power spectrum representations to determine a cutoff distance for judging two frameworks as identical. A histogram of these distances is provided in the Appendix, from which we conclude that structures within a distance of 5×10^{-6} from one another in SOAP space can be considered identical.

As a consequence of the class imbalances in our selected train set, which contains approximately 100 IZA frameworks and 10,000 Deem frameworks, the SVM is implicitly biased towards classifying samples as "Deem": for instance, the model can achieve 99% accuracy on the train set by trivially predicting all of the samples as Deem. To address this issue, we employ class weighting so that the SVM regularization is defined class-wise, where class k has regularization $C_k = C n_{\text{samples}}/(n_c n_k)$, where n_{samples} is the total number of samples, n_c the number of classes, n_k the number of samples belonging to class k, and C is a hyperparameter we optimize through a stratified two-fold cross-validated grid search to maximize the class-balanced accuracy on the validation set. For consistency with the class weighting, the centering of the input SOAP features (relative to the train set) is performed using a weighted mean, where all of the samples of a given class are weighted by $n_{\text{samples}}/(n_c n_k)$ and subsequently normalized such that the sum of all sample weights is equal to one.

Once the SVM models are trained, they are evaluated on a held-out test set comprising the remaining ≈100 IZA frameworks and ≈320,000 Deem frameworks. For each model we compute the predicted classes for the test set as well as the decision functions, which provide

a measure of how close a given sample is to the separating hyperplane and can be interpreted as a way to quantify the confidence level of a particular prediction. A histogram of the two-class "IZA vs. Deem" decision functions is shown in Fig. 7.9(a) for the full 6.0 Å SOAP power spectrum, illustrating that the model is clearly able to distinguish the two classes, as evidenced by the separate peaks corresponding to the IZA and Deem structures. Of particular note is that the decision function values for the Deem frameworks appear normally distributed, likely as a consequence of the uniform coverage of the structure space as mentioned in Section 7.2.4. The individual bars of the histogram are colored according to whether the corresponding classifications are true positives (TP), true negatives (TN), false positives (FP), or false negatives (FN), with Deem serving as the positive class and IZA as the negative class. The performance of the classification can be quantified through a receiver operating characteristic (ROC) curve [72, 74], shown as the line in Fig. 7.9(b). The ROC curve tracks the rate of false positives $FPR = FP/(FP + TN)$ and the rate of true positives $TPR = TP/(TP + FN)$ as the decision boundary is swept through the decision space as illustrated by the green arrows. A perfect classifier has ($FPR = 0, TPR = 1$), so that the closer the area under the ROC curve (AUC) is to one, the more accurate the classifier. A random guess corresponds to an ROC curve for which $FPR = TPR$ at every point. In principle, the Pareto optimum of the ROC curve corresponds to the FPR and TPR of the classification model for which it is constructed, illustrated by the green dot in Fig. 7.9, with which we can also associate a *confusion matrix* that tallies the number of true/false positive/negative classifications. Following our assumption that the most synthesizable Deem frameworks will exhibit structural similarities with IZA frameworks, we can narrow the search space by examining those Deem structures that are misclassified, i.e., the false negatives. However, this is not a particularly effective approach on its own, as even though approximately 90% of the IZA and Deem structures are classified correctly, this still leaves several thousand misclassified Deem frameworks, far too many to be of much practical use for identifying synthesis candidates.

Assessment of Stability

Up to this point, we have only considered the structural similarity between the IZA and Deem frameworks as an indicator of synthesis potential. To additionally account for thermodynamic stability, we construct a convex hull in a latent space based on the SVM decision functions. As the latent space we use two-component PCovR projections of the IZA and Deem frameworks, using the SOAP feature vectors as the predictor data and the SVM decision function values as the prediction targets. We train the PCovR models using the same training data as the SVM models, again accounting for class imbalance, though in a slightly different manner than for the SVM. In the PCovR models, we handle class imbalance by replicating the minority class (IZA) samples to achieve (approximate) class parity. We have chosen this approach instead of undersampling the majority class (Deem) or creating synthetic minority class examples using a technique such as SMOTE [238], as undersampling to achieve class parity would reduce the training set to ≈200 structures in the two-class case, and render the problem intractable in the four-class case, as the IZA canton populations are themselves imbalanced with the least populated class in the train set including less than 20 frameworks. Creating

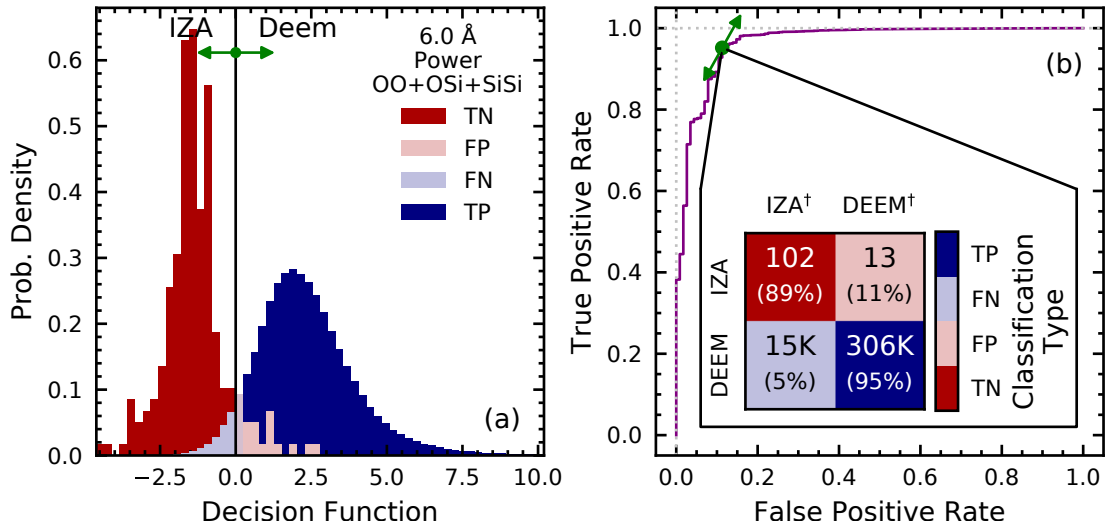

Figure 7.9 – (a) Histogram of decision function values and (b) corresponding ROC curve for the "IZA vs. Deem" SVM classification based on a 6.0 Å SOAP power spectrum representation as the decision function boundary is swept through the SOAP space. The inset of (b) also shows a confusion matrix for the two-class "IZA vs. Deem" classification using the full power spectrum SOAP vectors. The superscript † indicates predicted class labels.

synthetic IZA examples is also undesirable, as it distorts our baseline for the classification by introducing hypothetical frameworks into the "IZA" class, which is meant to exclusively contain experimentally synthesized structures. We optimize the PCovR mixing α and the regularization through a two-fold cross-validated grid search just as for the SVM (and using the same fold splitting), with the aim of minimizing the class-balanced PCovR loss (Eqn. 5.1) on the validation set, which does not contain replicated samples. Here again we preprocess the predictor and target data by centering relative to the column means of the train set and scaling by the Frobenius norm of the train set divided by the square root of the number of training samples.

The resulting PCovR latent space thus encodes the structural information of the IZA and Deem frameworks through (1) the raw SOAP features, and (2) the decision function values, which are proportional to the distances of the samples to the separating hyperplane in the SOAP feature space. In other words, the latent space arranges the frameworks along a mixture of the directions for which the variation in structural features as encoded by the SOAP vectors is the greatest, and the directions that correlate with synthesis conditions as encoded by the canton assignments.

To more robustly identify those frameworks from the Deem database that share structural similarities with IZA frameworks and that are more likely to be stabilizable, we apply a convex hull—similar in spirit to the generalized convex hull described in Section 5.4, but instead using a deterministic construction—to a two-component PCovR projection for our held-out test set of ≈ 100 IZA frameworks and $\approx 320{,}000$ Deem frameworks, using the GULP-calculated energies as the stability metric.

A visualization of the resulting convex hull construction is presented in Fig. 7.10(a), which shows the two-component PCovR projections of the IZA and Deem frameworks for which the four-class decision function values have been used as the property targets. Each framework is represented as a single point, colored according to its two-class "IZA vs. Deem" decision function value. The true IZA frameworks are represented as squares, and the true Deem frameworks are represented as circles. The size and transparency of the points indicate how close the corresponding framework is to the convex hull along the energy direction: larger, more opaque points lie closer to the hull, and the frameworks that serve as vertices for the convex hull are highlighted with thick black outlines. In this representation, the Deem frameworks that are most likely to be synthesizable are those that lie close to the hull and present as IZA (the large, opaque, red circles). Atomic snapshots are provided for five such structures that also have molar volumes greater than 60 $Å^3$/Si in addition to two IZA frameworks (SBN and MTN) for reference. Eighteen Deem frameworks serve as hull vertices and are also misclassified as IZA; these frameworks are thus the most promising Deem frameworks for experimental synthesis according to our methodology. We can expand our pool of synthesizable candidates by considering those frameworks, for example, within some cutoff distance from the hull along the energy axis, and having a decision function value less than some specified value. We can set these cutoff values in a number of ways, including basing them on the hull distances and decision functions of the IZA frameworks. For instance, there are approximately 11,700 Deem frameworks in the test set that are closer to the hull than the furthest all-silica IZA framework in the test set (6.18 kJ/mol Si) and have decision function values less than that of the test-set all-silica IZA structure that is the "most Deem" (decision function value 0.53). Since this is a rather large pool of structures, we choose to rank the frameworks, taking the top 50 Deem frameworks that are closest to the hull that are also misclassified as IZA (having decision function values < 0). These 50 structures are enumerated in Appendix C, where for each candidate we also provide the four-class cantonal predictions and the closest IZA framework in the SOAP feature space, which suggest the composition(s) at which each framework may be synthesizable. Synthesis candidates for a particular target application can be identified through a secondary, property-based filtering. For instance, if a more porous zeolite is desired, one can extract those zeolites possessing molar volumes above a certain threshold and subsequently rank them according to their distance from the hull and decision function value.

Figs. 7.10(b)–(d) give a high-level, statistical overview of the convex hull representation. Figs. 7.10(b) and (c) are histograms of the PCovR projections of the IZA and Deem frameworks, similar to Figs. 7.7(a)–(c), and show that the first component is organized according to the two-class "IZA vs. Deem" classification, while the second component is roughly organized according to the four-class cantonal classification. Fig. 7.10(d) shows a histogram of the distances between the frameworks and the final convex hull, indicating that the IZA frameworks largely appear close to the hull, much more so than the Deem frameworks. By virtue of the proximity of the (known to be synthesizable) IZA frameworks to the convex hull, this observation suggests that our convex hull construction in the PCovR latent space places those frameworks that are most likely to be stabilizable near the convex hull.

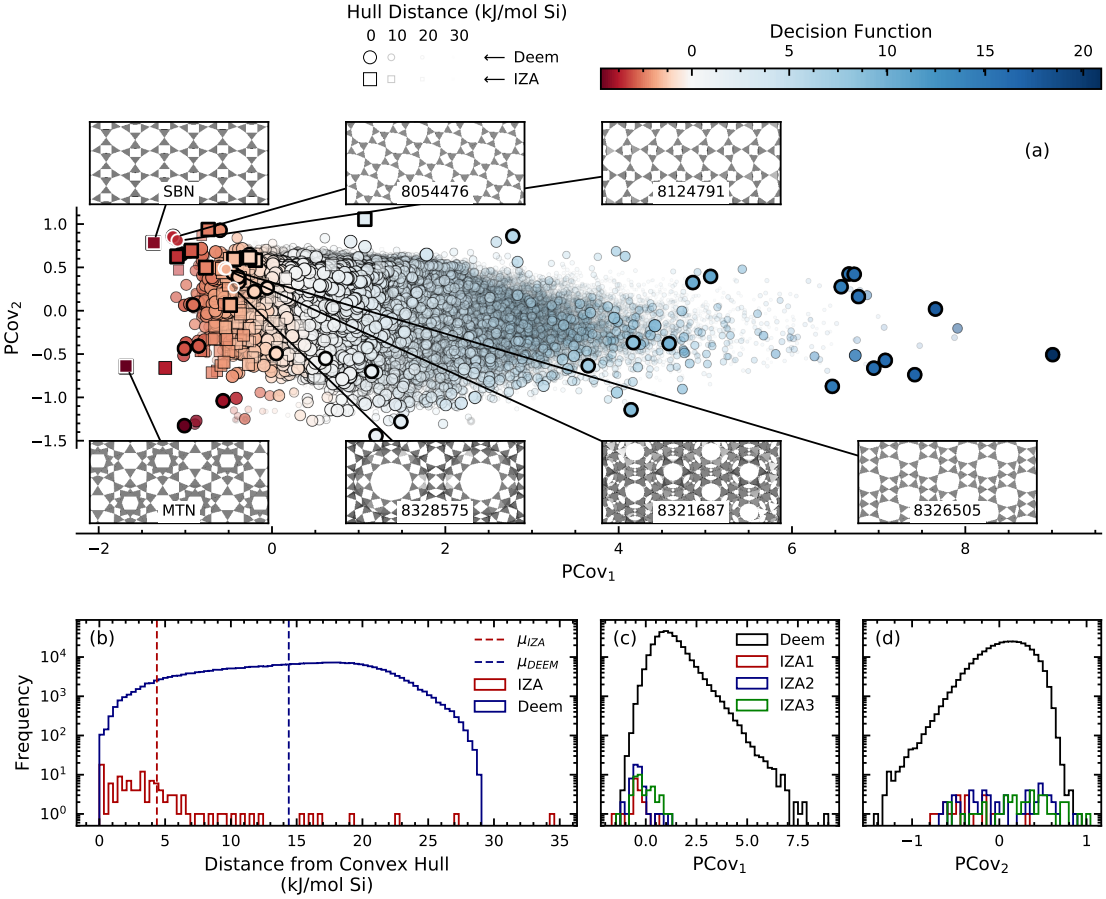

Figure 7.10 – (a) First two components of the PCovR projection based on the four-class cantonal decision functions with points colored according to the two-class IZA vs. Deem decision function. Each point represents a single framework and is sized and given an opacity according to its (energy) distance to the convex hull. Points become smaller and more transparent as their corresponding frameworks increase in distance to the hull. (b) Histogram of the energy distance to the convex hull for the IZA and Deem frameworks. (c)–(d) Histograms of the PCovR component values for the IZA cantons (excluding Canton 4, RWY) and Deem.

Table 7.2 – *AUC* for the two-class "IZA vs. Deem" SVM models based on the SOAP power spectrum.

	Power Spectrum						
	OO	OSi	SiSi	OO+OSi	OO+SiSi	OSi+SiSi	OO+OSi+SiSi
3.5 Å	0.931	0.943	0.940	0.937	0.958	0.941	0.950
6.0 Å	0.966	0.964	0.959	0.964	0.970	0.966	0.966

Table 7.3 – *AUC* for the two-class "IZA vs. Deem" SVM models based on the SOAP radial spectrum.

	Radial Spectrum		
	O	Si	O+Si
3.5 Å	0.930	0.862	0.932
6.0 Å	0.948	0.866	0.948

Analysis of Structural Features

Having identified several candidate Deem frameworks that might be experimentally synthesizable based on the convex hull, the question of *why* these structures appear to be synthesizable has yet to be explored. As the convex hull construction relies in large part on the prediction outcomes of the SVM model, we analyze the ensemble of classifiers described earlier in order to determine which structural features are most important for the decision-making process and thus for distinguishing the Deem frameworks from the IZA frameworks. As a first approach, we compare the performance of classifiers trained on SOAP representations with different cutoffs (3.5 Å and 6.0 Å) to determine the length scales of the most distinguishing features, on SOAP representations including only two-body or additionally three-body correlations to understand the required body order to make accurate classifications, and on subsets of the SOAP features corresponding to individual atomic correlations to find those that are the most different between the Deem and IZA frameworks. The ROC curves for the ensemble of these models for the "IZA vs. Deem" classification is shown in Fig. 7.11. Generally speaking, all of the models perform quite well, with only small differences in the associated *AUC* scores, which are given in Tables 7.2 and 7.3. The clear exceptions are the radial spectrum (two-body) models accounting for only Si–Si correlations, suggesting that information on the oxygen atoms is required in order to most accurately distinguish the Deem frameworks from the IZA. The 6.0 Å models also tend to perform better than their 3.5 Å counterparts, indicating that the inclusion of information past the first-neighbor shell can help fine-tune the classifications.

We can more intuitively visualize these results using confusion matrices, shown in Fig. 7.12 for the two-class case. (The four-class "IZA1 vs. IZA2 vs. IZA3 vs. Deem" case is provided in Appendix C.) The entries of the confusion matrix are colored according to the proportion of structures with a particular ground truth label (rows) having a particular predicted label (columns), with the interior text enumerating the absolute number of such (*true label, pre-*

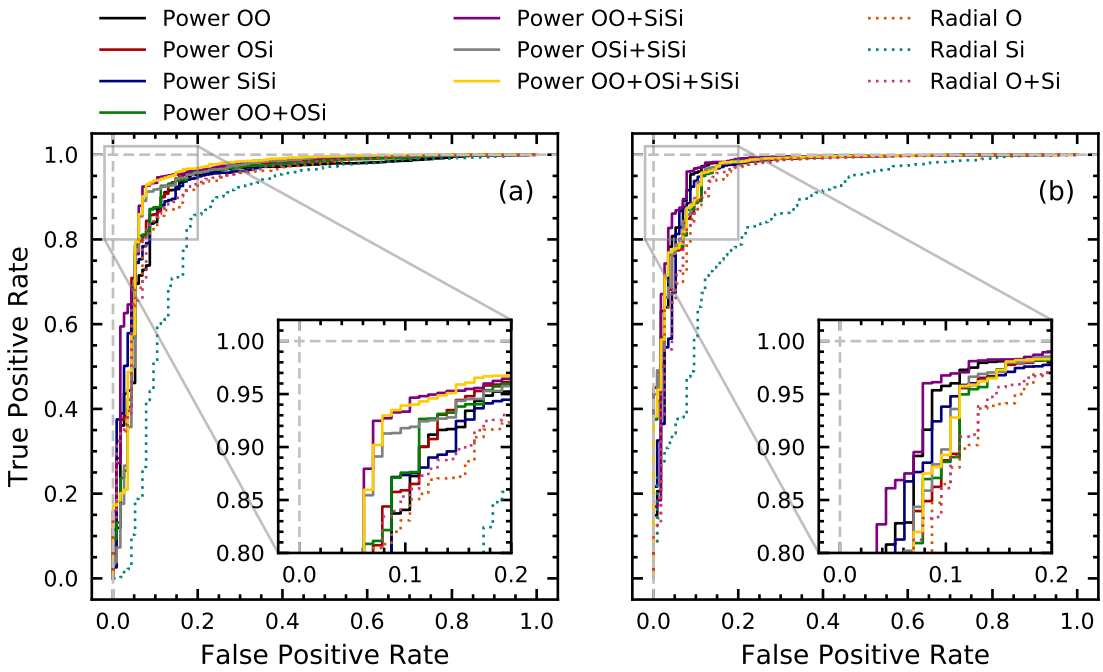

Figure 7.11 – Receiver operating characteristic (ROC) curves for the two-class "IZA vs. Deem" classification exercise where subsets of the power spectrum or radial spectrum were used for the SVM training and classification. The power spectrum features yield better predictions than the radial spectrum, with Si-O-Si correlations being particularly important for the classification. (a) provides the results for models using a 3.5 Å SOAP representation, while (b) gives results for the models based on a 6.0 Å SOAP representation.

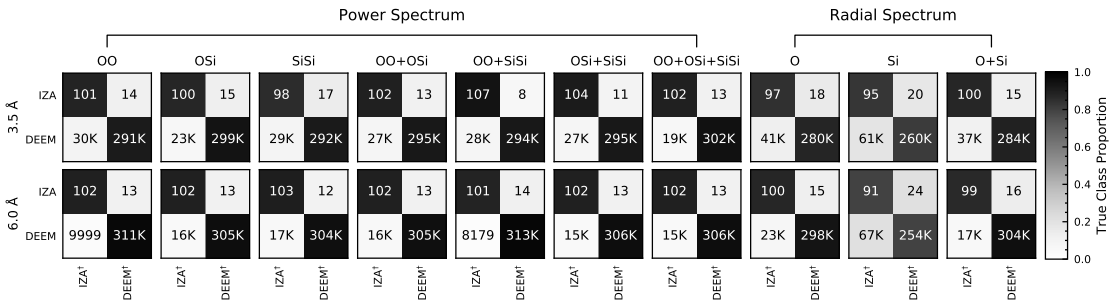

Figure 7.12 – Confusion matrices from the two-class SVM classifications where subsets of the power spectrum or radial spectrum were used for the SVM training and classification. The matrix entries are colored according to the proportion of the true labels that have been predicted as a particular class, while the interior text gives the absolute number of such classifications. The models correctly classify approximately the same number of IZA frameworks, differing mainly in the number of misclassified Deem frameworks. The superscript † indicates predicted class labels.

dicted label) pairs. With the exception of the Si-only radial models, all of the classifiers tend to correctly classify around 100 of the 115 IZA frameworks in the test set; where the models differ most substantially is in the number of misclassified Deem frameworks they produce, ranging from more than 60,000 for the 6.0 Å Si-only radial model to slightly more than 8,000 for the 6.0 Å power spectrum model including Si–O–O and Si–Si–Si correlations. To ensure that the classification predictions are due to genuine structural differences between the IZA and Deem frameworks, we also build SVM models trained on the Deem frameworks in our train set but with the class labels assigned at random. We find that the SVM is unable to predict the arbitrarily assigned labels, suggesting that the classification behavior we see is due to genuine structural differences between the frameworks. The confusion matrices for these "dummy models" are given in Appendix C.

While the analysis of the ROC curves and confusion matrices provide support for making general conclusions about the structural features in the IZA and Deem frameworks that are the most important for making classification decision, they do not provide the atomic-level resolution that would be most useful in developing an intuitive understanding of the similarities and differences between the hypothetical and synthesized frameworks. To develop this intuition, we construct real-space representations of the SVM model weights and the SOAP feature vectors as outlined at the end of Section 2.1.2, so that we can determine which physical features of the frameworks are most important to the SVM decision-making process. For this exercise we restrict our analysis to the radial spectrum models; even though they do not perform as well as the power spectrum in classifying the frameworks, they still result in reasonably accurate models while being conceptually simpler to visualize and understand, as the real-space radial density encoded by the SOAP vectors is simply the radial density function. Representations of the SVM "brains" for the radial spectrum models are thus shown in Fig. 7.13, where the subplot columns correspond to the 3.5 Å (left) and 6.0 Å (right) radial spectrum models, and the rows correspond, top to bottom, to SVM models trained on only

Si – O correlations, Si – Si correlations, and both Si – O and Si – Si correlations. The latter is split into two rows, where the top shows only the Si – O correlation contributions and the bottom shows only the Si – Si contributions. In each panel, the class-weighted average radial SOAP density (of the frameworks in the train set) $\overline{\overline{\rho}}(r)$ is plotted alongside the "cumulative decision function" for 50 randomly selected frameworks (25 IZA, 25 Deem) from the test set. The background of each panel is colored according to the real-space representation of the SVM weights $w(r)$. For a given framework with radial density $\rho(r)$, the SVM makes a classification based on an integral of the product $w(r)\Delta\rho(r)$ plus a bias term b, where $\Delta\rho(r) = \rho(r) - \overline{\overline{\rho}}(r)$. Hence, the cumulative decision function $F(r)$ can be defined as

$$F(r) = b + \int_0^r \mathrm{d}r' \, w(r')\Delta\rho(r'), \tag{7.3}$$

and shows what the classification decision would be if the representation was truncated at a distance r. $F(r_c^+)$, where r_c^+ is distance at which the radial density of the representation converges to zero past the SOAP environment cutoff, is thus the final decision function value. Note, however, that $F(r)$ is merely a representation of the SVM decision-making process and does not depict how the SVM actually makes the classification decisions for the zeolite frameworks. Nonetheless, the cumulative decision function serves as a more intuitive representation for highlighting the structural features that are the most important to the decision-making process. More specifically, the distance at which the cumulative decision functions for IZA $F(r)_{IZA}$ and Deem $F(r)_{Deem}$ diverge most significantly is the length scale of the structural features that best distinguish the two classes. Consider the example of the 6.0 Å radial model trained only on Si – O correlations, shown in Fig. 7.13(e). There is a clear divergence of the $F(r)$ for IZA and Deem at $r \approx 3$ Å, corresponding to the onset of the density from the second-neighbor O atoms, suggesting that these are the correlations most influential in the "IZA vs. Deem" classifications. In other words, the SVM model picks up on small differences in the tails of the density of the second-neighbor O atoms, and uses this information as the main determining factor in the classification. We can understand how these density differences affect the decision by examining the SVM weights at the location of the $F(r)$ divergence. Since the decision function is based on the product $w(r)\Delta\rho(r)$, the background weights can be interpreted as a "gradient" on which the density lies. More specifically, since $F(r_c^+) > 0$ defines a "Deem" classification and $F(r_c^+) < 0$ defines an "IZA" classification, if at a given r the product $w(r)\Delta\rho(r)$ is positive, the decision moves towards "Deem"; if the product is negative, the decision moves towards "IZA". With this in mind, we observe in Fig. 7.13(e) that at the divergence of the $F(r)$ at $r \approx 3$ Å, the weights are positive; thus, if a framework exhibits a depletion of density in the region ($\Delta\rho(r) < 0$), it will likely be predicted as "IZA". From a structural perspective, then, if we wish to find Deem frameworks with IZA-like structural features, we should search for frameworks with slightly larger average second-neighbor Si – O distances.

Another interesting observation that can be drawn from the cumulative decision functions is the negligible contribution that Si – Si correlations give when Si – O correlations are available. To arrive at this conclusion we examine the bottom two rows of Fig. 7.13, which show the cumulative decision functions for the 3.5 Å and 6.0 Å models trained on both Si – O

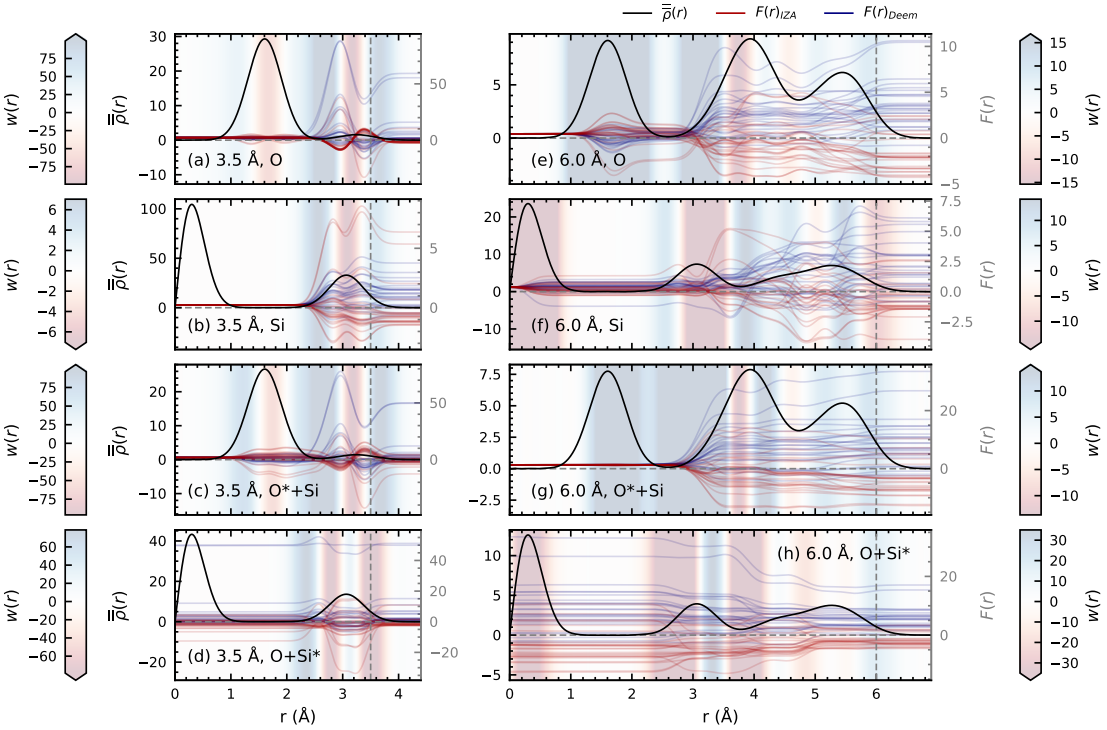

Figure 7.13 – The class-averaged SOAP-reconstructed radial atom density $\overline{\overline{\rho}}(r) = \frac{1}{2}\left(\overline{\rho}(r)_{Deem} + \overline{\rho}(r)_{IZA}\right)$ is plotted alongside the cumulative decision function $F(r)$ for 25 random IZA and 25 random Deem frameworks. The plot background is colored according the the value of the SVM weights $w(r)$, where the large magnitude weights have been saturated in color to more clearly show sign changes. The environment cutoff for the SOAP representation is indicated by the vertical dashed line; the SVM decision boundary $F(r_c^+) = 0$ is given by the horizontal dashed line. For (c)–(d) and (g)–(h), the subplots are labeled to indicate the relevant contributions in models based on multiple correlations. For instance, the label "O*+Si" denotes the Si – O correlation contributions to the classification decisions of an SVM model based on both Si – O and Si – Si correlations.

and Si – Si correlations; the top panels show contributions from only Si – O correlations while the bottom panels show contributions from only Si – Si correlations. As such, the value of $F(r_c^+)$ for the Si – O correlations (Figs. 7.13(c) and 7.13(g)) are not the final decision function values; instead, the value of $F(r_c^+)$ for the Si – O correlations is carried over to $F(0)$ in the Si – Si correlations, so that the values of $F(r_c^+)$ in the Si – Si correlations (Figs. 7.13(d) and 7.13(h)) are the final decision function values for the combined (Si – O)+(Si – Si) correlation models. By representing the models in this way, the individual correlation contributions to the final decision function become clear. Notably, $F(r)$ is rather flat in the Si – Si correlation models and sign changes are rare. Consequently, we can conclude that Si – Si correlations play a negligible role in the decision-making process when Si – O correlations are available.

Taking together the the accuracy metrics (ROC curves and confusion matrices) of the SVM model ensemble and our analysis of the SVM "brain" for the radial spectrum, we can conclude

that information on the O atoms is essential for distinguishing the IZA frameworks from the Deem frameworks, with the most important features for making the distinction being the low-r density tail of the second-neighbor O atoms. While two-body correlations are sufficient to distinguish IZA from Deem with 80–90% class-balanced accuracy, The inclusion of three-body correlations further improves the resolving power of the SVM models, as does increasing the length scale of the correlations that are included in the local atomic environments.

7.4 Conclusions

By building upon the concepts introduced in Chapter 6, we have developed here an approach for visualizing local atomic environments in zeolite frameworks in a way that transparently encodes structure–property relationships. The resulting map is based on a KPCA latent space of atom-centered SOAP environments to which we attributed molar volume and energy contributions through supervised machine learning. We extended this mapping approach to the materials discovery task of identifying potentially synthesizable frameworks contained in a database of hypothetical all-silica zeolite structures. To this end, we augmented the latent space through PCovR, using predictions from supervised classification models to emphasize the directions in the feature space most relevant for distinguishing the hypothetical frameworks from those that have been synthesized. We subsequently applied a convex hull construction to the PCovR latent space to identify the frameworks that are the most thermodynamically stable. As a result of this procedure, we could straightforwardly identify the most promising hypothetical frameworks for experimental synthesis as those that lay close to the hull and were misclassified as synthesized frameworks. We concluded by examining the decision-making process of the classification models to understand which particular structural features are most responsible for distinguishing the hypothetical frameworks from the synthesizable. Altogether, this sequential workflow serves as a demonstration of how supervised and unsupervised machine learning can be combined in sophisticated ways to uncover structure–property relationships and to perform materials discovery tasks that yield more robust results than supervised or unsupervised learning alone. Since its constituent techniques are application agnostic, the workflow is also adaptable to different materials systems, particularly where there exist sufficient data to identify patterns in properties and structural characteristics between known and hypothetical materials.

8 Conclusions

Over the past several years, machine learning methods have become common analysis tools in chemistry and materials science. While most applications of machine learning in these fields tend to apply supervised and unsupervised techniques in isolation, this thesis takes a different perspective by considering the added benefit of combining both supervised and unsupervised learning in a variety of ways, ranging from the simple, longstanding approach of using unsupervised dimensionality reduction methods as a preprocessing step for supervised learning, to the construction of more complex workflows and the examination of correlations between unsupervised representations and supervised predictions.

After showing that unsupervised clustering techniques can provide a means for constructing transferable definitions of hydrogen bonding motifs in protein crystal structures, we applied the same clustering methodology to examine patterns in dihedral angle sequences and local atomic environments along the protein backbone. While we found moderate correspondence between the data-driven motifs and the most well-defined secondary structures (α-helices and β-strands), we found a general lack of alignment between the motifs and the secondary structure assignments at large. Through the use of support vector classification, we were able to show that the lack of correspondence between the identified motifs and conventional secondary structure definitions were not a result of deficiencies in the feature representation, but because the conventional, heuristic-based classifications are not wholly reflected in the statistical distribution of structural features of the protein structures.

As an additional example of combined supervised–unsupervised machine learning, we built a map of local, atom-centered structure–property relationships in a collection of hypothetical zeolite frameworks. We constructed this map by using kernel ridge regression to predict the molar volumes and energies of the frameworks and subsequently decomposing the predictions into contributions from the individual environments composing the structure. As the "coordinate system" of the map we used the first few principal components of the environment feature vectors, as they were shown to correlate with the volume and energy contributions. The resulting map thus naturally orders the zeolite environments by their similarity in structure and properties. A map exhibiting even stronger structure–property connections can be constructed through methods like (kernel) principal covariates regression, which yield tunable low-dimensional latent spaces that explicitly include contributions from the feature space representation and target predictions.

Having shown that machine learning can be used to compare local structure in hypothetical zeolites, we set out to make comparisons between hypothetical frameworks and frameworks that have been experimentally synthesized with the ultimate aim of finding the needle-in-a-haystack hypothetical structures that are most promising for experimental synthesis. To this end we once again employed support vector classification, this time to distinguish the experimental frameworks from the hypothetical. The support vector machine decision function values, which serve as a measure of the confidence of the corresponding classification predictions, were used to construct a principal covariates regression latent space on which a convex hull construction was applied to identify the most stabilizable structures. Through this workflow chaining supervised, unsupervised, and hybrid methods, we were afforded several criteria with which we could identity the most suitable candidates for experimental synthesis among the hypothetical zeolite frameworks.

Through these examples, we have shown that combining both supervised and unsupervised machine learning through workflows and hybrid models can help to form a more complete picture of structural motifs, materials properties, and the connections between them in large databases of complex materials. Consequently, the work presented here suggests that more widespread adoption of methodologies that utilize multiple paradigms of machine learning will permit deeper understanding of complex relationships and processes in and among materials and molecules, facilitating the discovery of novel structures and properties and further driving technological innovation.

A Preprocessing and Model Tuning

A.1 Model Construction

Machine learning models are constructed by exposing the model to a set of example (*input*, *output*) pairs that serve to condition, or *train* the model to predict the outputs for inputs it has never seen before. To this end, when applying a machine learning technique to a particular set of data, the data is typically split into a training set and a test set. The training set is used to condition the model and to optimize any hyperparameters (see section A.1.1), while the test set is used to evaluate the model and quantify how well the model is expected to perform on unseen data. It is advisable to keep the training and test sets completely separate, as incorporating some (or all) of the test set samples, features, or other characteristics (such as the mean) into the training data allows information from the test set to "leak" into the model so that it can "cheat" and give an overly optimistic assessment of its predicted performance on truly novel data, since it has information about the overall collection of data rather than that confined to the train set [72–74, 239].

A.1.1 Cross Validation

Many machine learning models include—either explicitly or implicitly—tunable parameters, such as the regularization or kernel width. It is common practice to determine these hyperparameters by constructing an ensemble of models using different combinations of hyperparameters, selecting as the optimal parameters those corresponding to the best-performing model based on some metric. If the data are plentiful, a subset of the training data, often called the *validation set*, can be set aside for evaluating models during hyperparameter selection. In such cases, the ensemble of models is fitted on the remaining training data, and the performance of each model is evaluated on the validation set. The performance of the model with the optimal hyperparameters can then be trained on the samples in the combined training and validation sets and assessed based on the held-out test [72]. If, however, the dataset is small enough to make a three-way split of the samples impractical, the optimization of the hyperparameters can be performed using a different approach known as *cross validation*, which is a useful method for optimizing hyperparameters with an efficient use of the available data. In k-fold cross validation [72–74], one of the most popular approaches, the training set is divided in to k equally sized (to the extent possible), non-overlapping "folds" . k models are then constructed, each one tested on a different fold and trained on the remaining $k-1$ folds.

This makes it possible to optimize the hyperparameters in a way that increases the likelihood that the resulting hyperparameters will generalize well to unseen data. When k is equal to the number of samples in the training set, this is known as leave-one-out cross validation. Data preprocessing should generally be carried out separately within each of the k folds to ensure that the model has access to and is trained on *exclusively* the information in its $k-1$ training folds. The samples belonging to each fold can be selected at random, or they can be constructed using a stratification procedure. In the classification context, this means that the folds are selected such that each one contains approximately the same class proportions as the full training set [72]. In the regression context, stratification entails dividing the sorted target properties into $n_{samples}/k$ "buckets". Each fold then contains one sample from each "bucket" [4].

A.1.2 Centering and Scaling

In both linear and kernel methods, it is occasionally recommended (or required) to center and scale the feature data before training the machine learning model. However, how the centering and scaling is carried out depends on the particular model and feature representation. Centering the feature data in linear methods involves subtracting the feature values averaged over the training set from the samples in both the training and test sets. In regression models, a similar centering is performed for the targets. Scaling the features can either be performed on globally or on a per-feature basis, with the most appropriate choice being dependent on the particular feature representation. If the features are unrelated, perhaps possessing very different scales, scaling features individually is advisable; however, if the features are related and the relative magnitudes of the features carry information about the sample, then any scaling should be applied globally. Scaling is typically done by dividing the (centered) features in the test and train sets by the Frobenius norm of the feature matrix $\mathbf{X}$ of the training set or by a measure of the variance of $\mathbf{X}$, e.g., the trace of the covariance or the featurewise variances. The prediction targets should be scaled using the same rationale. Centering and scaling is of particular importance in PCovR-based methods, as imbalanced scaling of the features and targets can bias the model.

In kernel methods, an appropriately centered and scaled kernel is defined as the dot product of the centered and scaled RKHS features. However, we can obtain a properly preprocessed kernel matrix without explicitly computing the RKHS features and instead act directly on the kernel. The kernel centering operation can be expressed for the $N' \times N$ kernel matrix $\mathbf{K}'$ as $\mathbf{K}' - \overline{\mathbf{K}}$, where the "kernel mean" $\overline{\mathbf{K}}$ is [126],

$$\overline{\mathbf{K}} = \mathbf{1}_{N'N}\mathbf{K} + \mathbf{K}'\mathbf{1}_{NN} - \mathbf{1}_{N'N}\mathbf{K}\mathbf{1}_{NN} \tag{A.1}$$

where $\mathbf{1}_{N'N}$ is an $N' \times N$ matrix with each element equal to 1/N, and $\mathbf{K}$ is the $N \times N$ kernel matrix between the samples in the training set. Normalizing the RKHS features $\mathbf{\Phi}$ by their Frobenius norm is equivalent to normalizing the kernel by its trace.

When using approximate RKHS features from a Nyström approximation, the centering and scaling takes a form analogous to the centering of the full kernel, but instead the approxi-

mation to the RKHS is centered. Equivalently, the kernel between a set of input samples and the representative samples $\mathbf{K}'_{NM}$ is centered and scaled such that the low-rank approximation of the full kernel is appropriately preprocessed. This amounts to subtracting $\mathbf{1}_{NN}\mathbf{K}_{NM}$ from $\mathbf{K}'_{NM}$ (i.e., subtracting the column means on $\mathbf{K}_{NM}$), with $\mathbf{K}_{NM}$ being the kernel between the training set samples and the representative samples. Normalizing the approximate RKHS features by their Frobenius norm is equivalent to dividing $\mathbf{K}'_{NM}$ by $\sqrt{\mathrm{Tr}\left(\mathbf{K}_{NM}\mathbf{K}_{MM}^{-1}\mathbf{K}_{NM}^{T}\right)}$ after $\mathbf{K}_{NM}$ has been centered.

B Protein Secondary Structures [1]

B.1 Probability Distributions

Fig. B.1 is the STRIDE analog to the DSSP dihedral angle probability distribution presented in the Section 6.3.5. Figs. B.2–B.5 are the DSSP and STRIDE probability distributions in six and ten dimensions. The higher dimensional dihedral angle spaces are formed by considering the dihedrals from consecutive residues. In all cases, the helices and strands are represented primarily by one or two clusters, while the other secondary structures tend to be spread across several clusters.

Fig. B.6 is the STRIDE analog to the DSSP SOAP probability distribution presented in Section 6.3.5. Figs. B.7–B.10 are the DSSP and STRIDE probability distributions in six and ten dimensions. The higher dimensional SOAP spaces are formed by considering additional principal components of the collection of SOAP vectors after reducing the number of features via farthest point selection. In contrast to the dihedral angle representations, clustering based on the SOAP representation does not result in the strands and helices being clearly confined to one or two clusters, particularly in the two- and six-dimensional cases.

B.2 Supervised Classification

Table B.1 provides the computed Q3 and Q8 scores for the dihedral angle and SOAP representations associated with the STRIDE secondary structure classification, similar to Table 6.3 in Section 6.3.6 that uses the DSSP assignments. Fig. B.11 shows the learning curves of the Q3 and Q8 scores for the support vector classification of the STRIDE secondary structure labels.

[1] This appendix is adapted with modifications under the Creative Commons Attribution 4.0 (CC BY 4.0) license from the supplementary material of Helfrecht, B. A., Gasparotto, P., Giberti, F. & Ceriotti, M. Atomic Motif Recognition in (Bio)Polymers: Benchmarks From the Protein Data Bank. *Frontiers in Molecular Biosciences* **6,** 24. doi:10.3389/fmolb.2019.00024 (2019); BAH performed the data analysis and prepared figures, PG ran preliminary tests, and all authors contributed to the design of the study and to the writing of the manuscript from which the present text has been adapted.

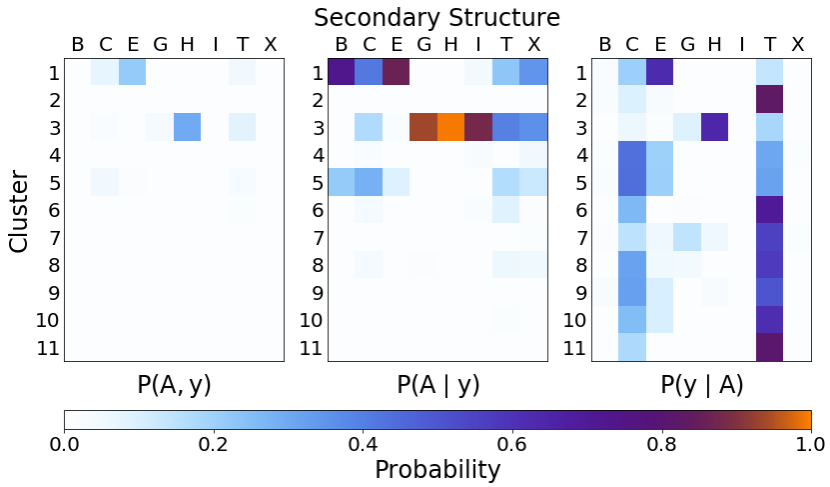

Figure B.1 – Joint and conditional probabilities for the secondary structures obtained from STRIDE and the clustering of dihedral angles from PAMM, where A is the cluster assignment and y the secondary structure classification. Reproduced from Ref. [160] under CC BY 4.0.

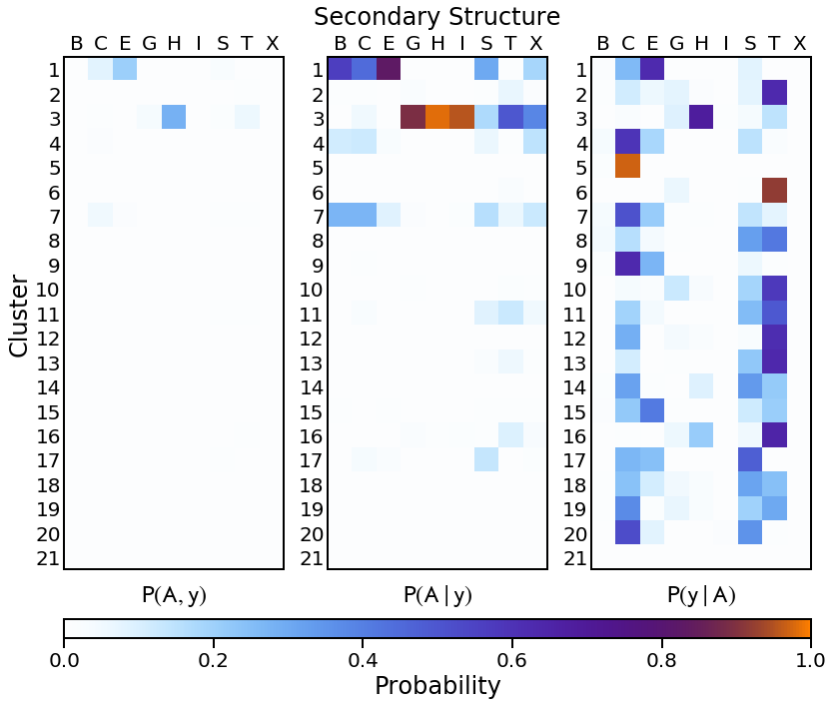

Figure B.2 – Joint and conditional probabilities for the clustering of dihedral angles from PAMM for three consecutive residues (a six-dimensional ϕ, ψ space), where A is the PAMM cluster assignment and y is the DSSP secondary structure assignment of the middle residue. Reproduced from Ref. [160] under CC BY 4.0.

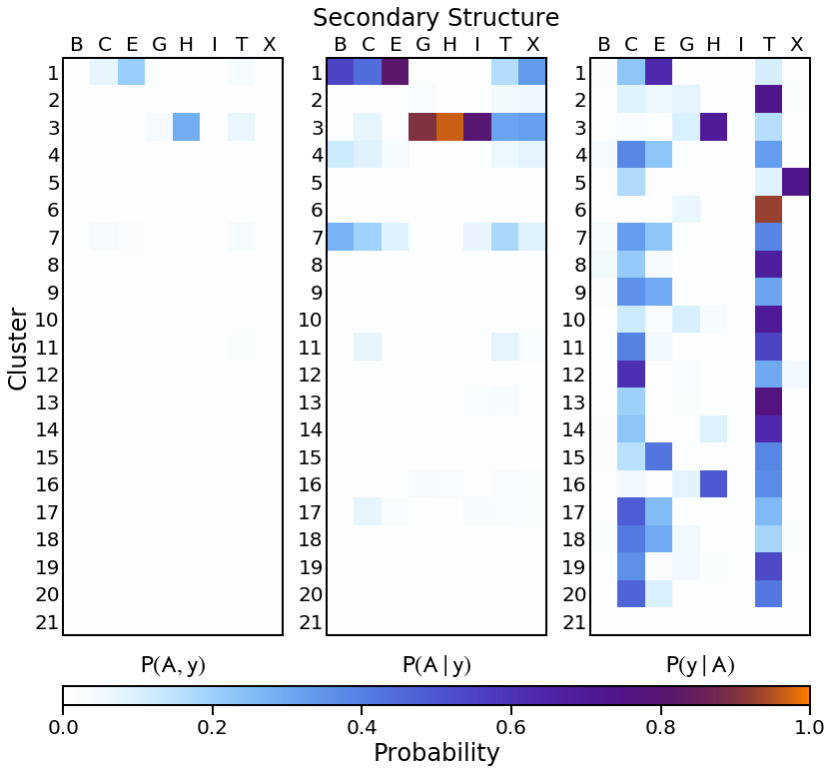

Figure B.3 – Joint and conditional probabilities for the clustering of dihedral angles from PAMM for three consecutive residues (a six-dimensional ϕ, ψ space), where A is the PAMM cluster assignment and y is the STRIDE secondary structure assignment for the middle residue. Reproduced from Ref. [160] under CC BY 4.0.

Table B.1 – Q3 and Q8 scores relative to STRIDE for PAMM PMI and SVM predictions of secondary structure based on a PCA of SOAP vectors and dihedral angles at various dimensionality. The reported SVM scores are an average over five separate constructions of the SVM, each time using a new random subset of 200,000 residues, with 50,000 of these serving as the training set.

	PAMM PMI		SVM	
Representation	Q3	Q8	Q3	Q8
ϕ, ψ (2D)	0.72	0.61	0.77	0.65
ϕ, ψ (6D)	0.74	0.62	0.86	0.76
ϕ, ψ (10D)	0.73	0.62	0.89	0.81
SOAP PCA (2D)	0.74	0.60	0.76	0.65
SOAP PCA (6D)	0.72	0.60	0.85	0.75
SOAP PCA (10D)	0.71	0.58	0.90	0.80
SOAP PCA (100D)	—	—	0.95	0.88

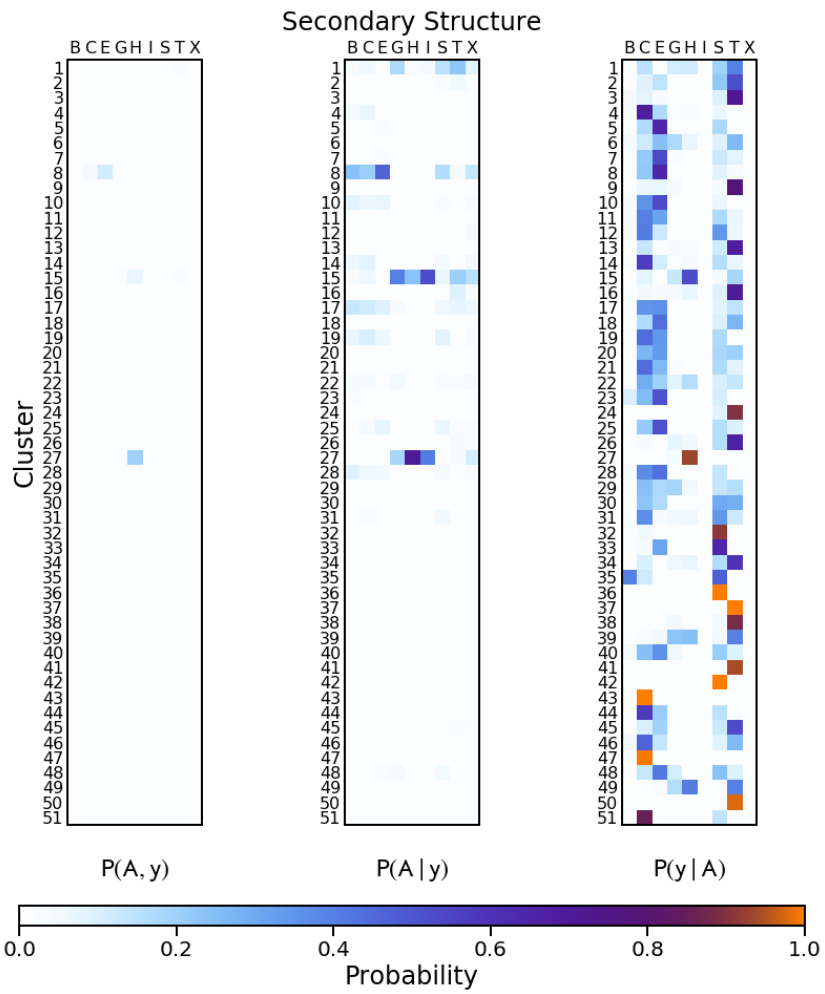

Figure B.4 – Joint and conditional probabilities for the clustering of dihedral angles from PAMM for five consecutive residues (a ten-dimensional ϕ, ψ space), where A is the PAMM cluster assignment and y is the DSSP secondary structure assignment of the middle residue. Reproduced from Ref. [160] under CC BY 4.0.

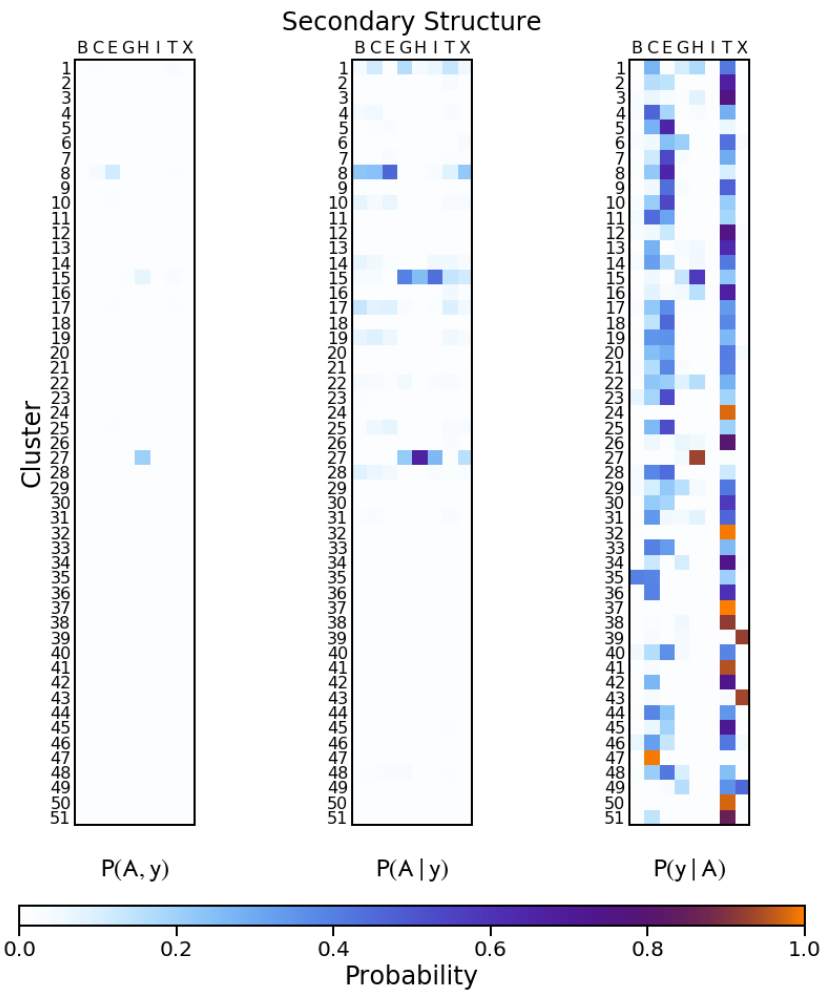

Figure B.5 – Joint and conditional probabilities for the clustering of dihedral angles from PAMM for five consecutive residues (a ten-dimensional ϕ, ψ space), where A is the PAMM cluster assignment and y is the STRIDE secondary structure assignment of the middle residue. Reproduced from Ref. [160] under CC BY 4.0.

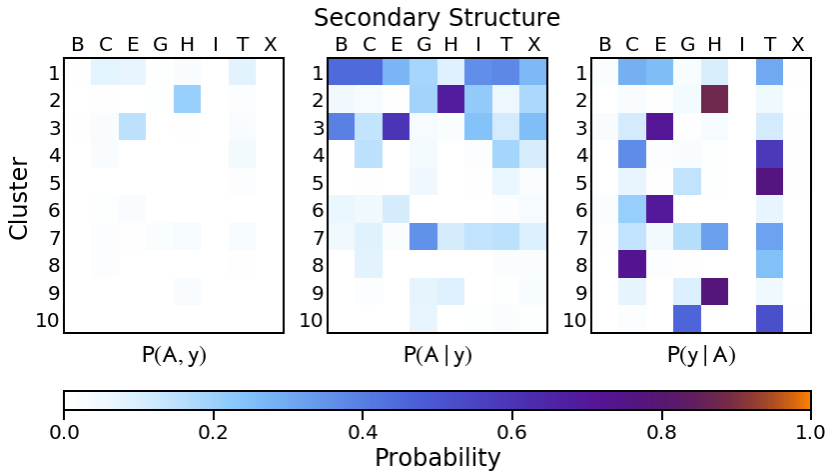

Figure B.6 – Joint and conditional probabilities for the PAMM clustering of the first two principal components of the reduced SOAP vectors describing each residue of the protein backbone, where A is the PAMM cluster assignment and y is the STRIDE secondary structure classification. Reproduced from Ref. [160] under CC BY 4.0.

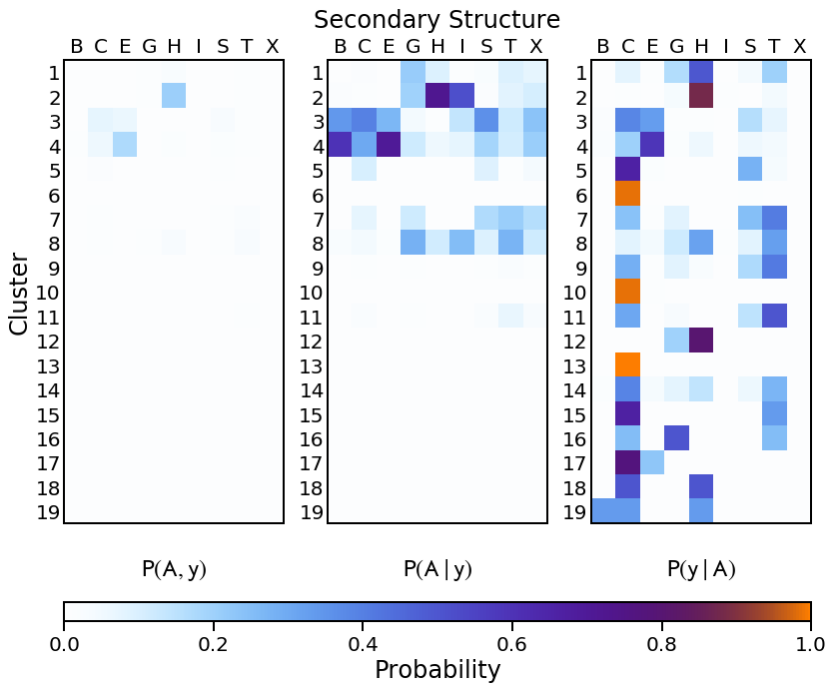

Figure B.7 – Joint and conditional probabilities for the PAMM clustering of the first six principal components of the reduced SOAP vectors describing each residue of the protein backbone, where A is the PAMM cluster assignment and y is the DSSP secondary structure classification. Reproduced from Ref. [160] under CC BY 4.0.

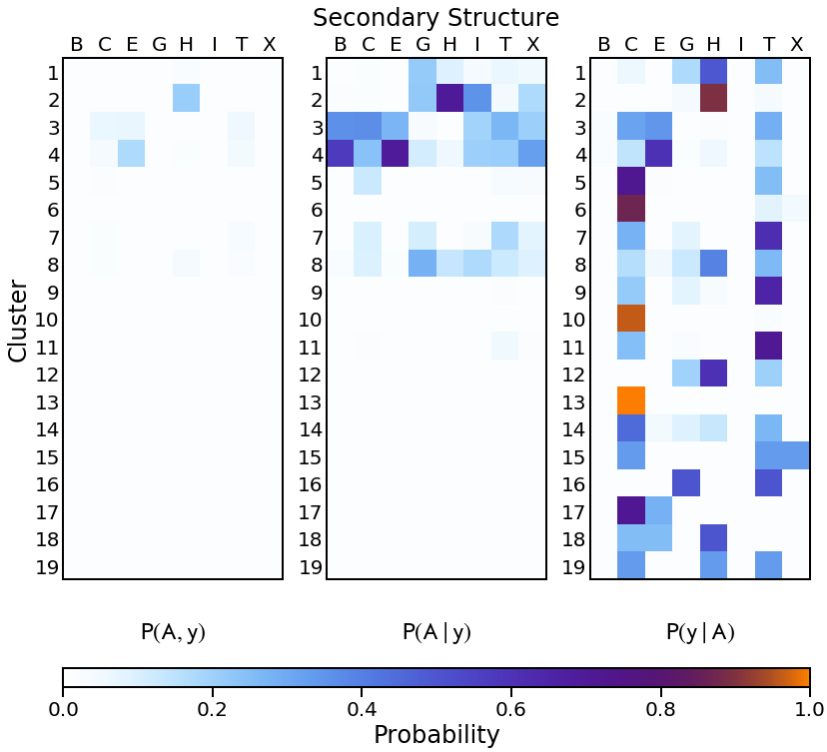

Figure B.8 – Joint and conditional probabilities for the PAMM clustering of the first six principal components of the reduced SOAP vectors describing each residue of the protein backbone, where A is the PAMM cluster assignment and y is the STRIDE secondary structure classification. Reproduced from Ref. [160] under CC BY 4.0.

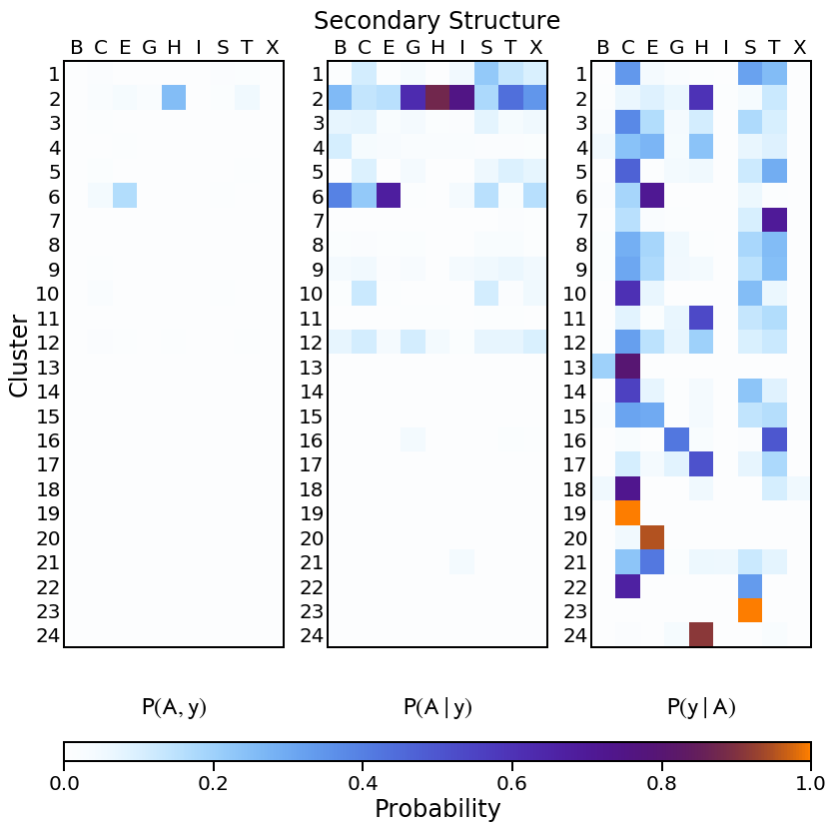

Figure B.9 – Joint and conditional probabilities for the PAMM clustering of the first ten principal components of the reduced SOAP vectors describing each residue of the protein backbone, where A is the PAMM cluster assignment and y is the DSSP secondary structure classification. Reproduced from Ref. [160] under CC BY 4.0.

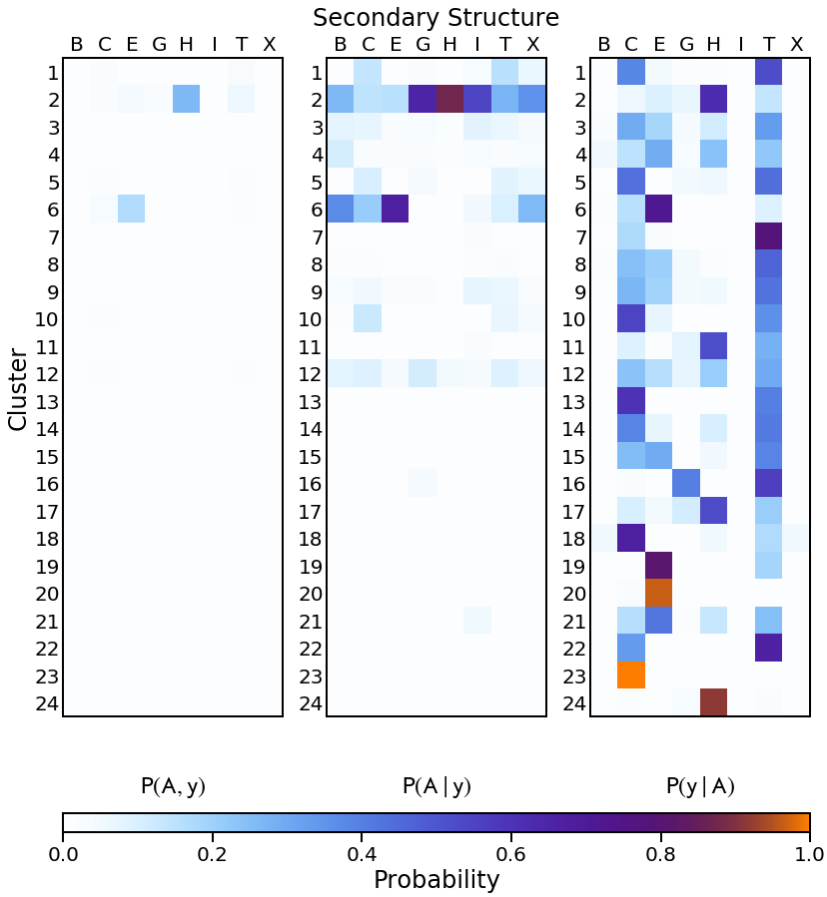

Figure B.10 – Joint and conditional probabilities for the PAMM clustering of the first ten principal components of the reduced SOAP vectors describing each residue of the protein backbone, where A is the PAMM cluster assignment and y is the STRIDE secondary structure classification. Reproduced from Ref. [160] under CC BY 4.0.

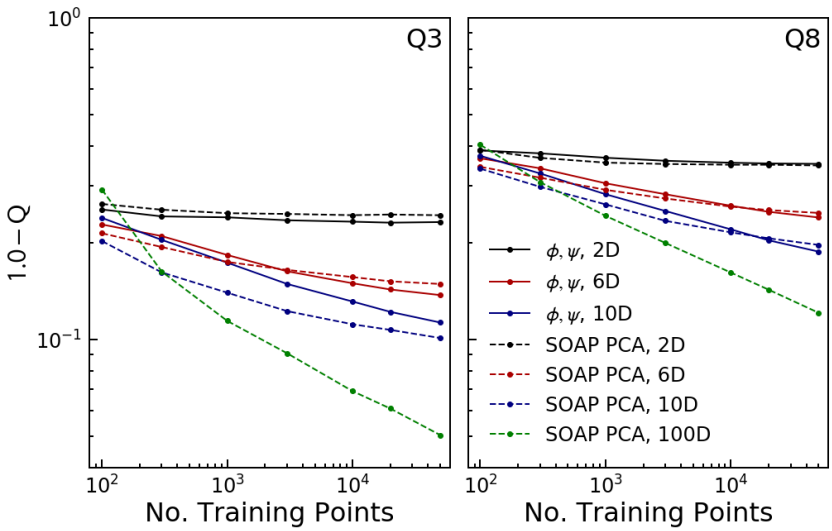

Figure B.11 – Learning curves of Q3 and Q8 scores relative to STRIDE for the multiclass SVM based on backbone dihedral angles and a PCA of the SOAP representation with various degrees of information content (i.e., the dimensonality of the descriptor). The Q scores are represented in the learning curves as errors, i.e., $1 - Q$. Reproduced from Ref. [160] under CC BY 4.0.

C Zeolites [1]

C.1 Ring-Based Descriptors

In terms of predicting the molar volume and energy, both King's definition and the shortest path definition perform very similarly. Fig. C.1 show the learning curves for the ring descriptors built on the 1,000- and 10,000-structure samples. Two variations of the rings descriptor are examined. The "Distribution" ("Dist.") variant is the descriptor described in Chapter 7: the sth element of the feature vector for a given Si-centered environment is the number of rings of size s that include the central Si. The "Binary" ("Bin.") variant is a binary version of the "Distribution" representation: the sth element of the feature vector is 1 if the central Si participates in at least one ring of size s and is 0 otherwise. As noted in Chapter 7, the FPS of the ring descriptor often results in fewer than 2,000 unique environments. In these cases, only the unique feature vectors serve as representatives in the learning models. Consequently, the models based on the "binary" variants of the King and shortest path ring descriptors use 109 and 53 representatives for the 1,000-structure sample, and 239 and 94 representatives for the 10,000-structure sample, respectively; the model for the "distribution" variant of the shortest path ring counts in the 1,000-structure sample uses 763 representatives. The "distribution" shortest path descriptor for the 10,000-structure set uses 2,000 unique representatives.

C.2 Results for the 1,000-Structure Subset

C.2.1 Learning Curves

Fig. C.2 shows the learning curves for the classical and SOAP descriptors for the 1,000-structure subset, analogous to Fig. 7.3. The learning curves of the 1,000-structure subset are similar to the results of the 10,000-structure subset for the first 1,000 training points.

[1] Sections C.1–C.3, and their corresponding subsections, of this appendix are adapted with modifications from the supplementary material of Helfrecht, B. A., Semino, R., Pireddu, G., Auerbach, S. M. & Ceriotti, M. A New Kind of Atlas of Zeolite Building Blocks. *The Journal of Chemical Physics* **151,** 154112. doi:10.1063/1.5119751 (2019), with the permission of AIP publishing; BAH performed the machine learning analyses and prepared the corresponding figures, RS and GP performed preliminary analyses and input processing and computed the classical descriptors, and all authors contributed to the design of the study and to the writing of the manuscript from which the present text has been adapted. Section C.4 contains work currently in preparation for submission; for this work, BAH performed the machine learning analyses and prepared figures, and all authors (Helfrecht, Semino, Pireddu, Auerbach & Ceriotti) contributed to the design of the study.

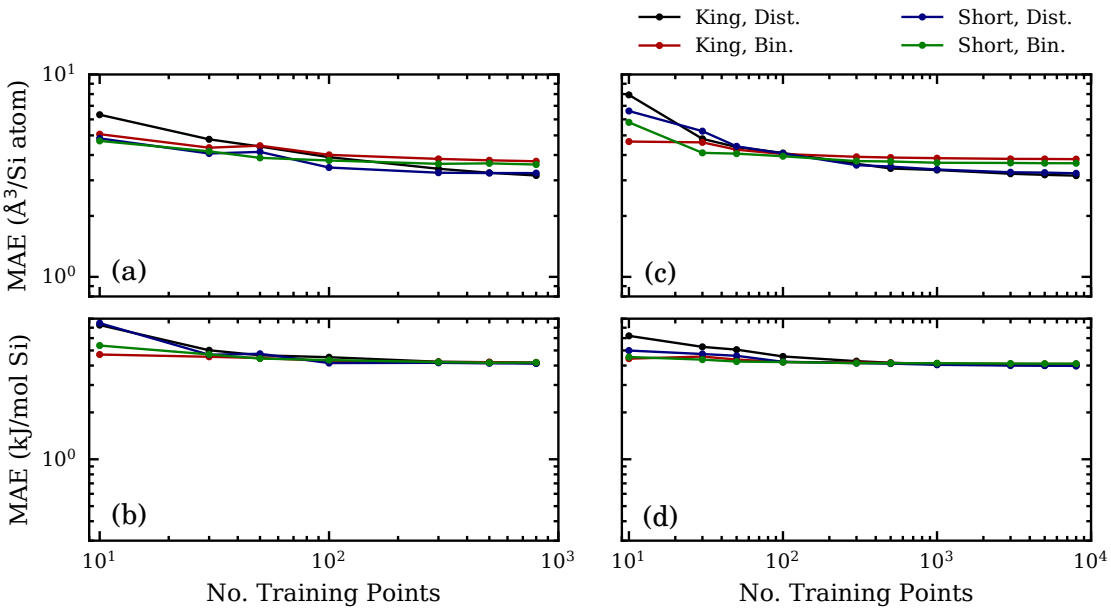

Figure C.1 – Learning curves for the ring-based descriptors from the sample of 1,000 structures for predicting the (a) volume per Si atom and (b) the energy per mol Si. (c)–(d) show the corresponding learning curves for the 10,000-structure subset. Adapted from Ref. [204] with permission of AIP publishing.

C.2.2 Property Correlations

Fig. C.3 shows the volume and energy correlations with the 3.5 Å and 6.0 Å SOAP-KPCA representations for the 1,000-structure subset.

C.3 Learning Curves on SOAP-KPCA Descriptors

Figs. C.4 and C.5, show the learning curves for the prediction of zeolite volume per Si atom and energy per mol Si using the SOAP-KPCA representation with different numbers of principal components. As the number of principal components composing the representation is increased, the prediction becomes more accurate.

In the case of predicting the volume per Si atom, a SOAP-KPCA representation including 50 principal components performs similarly to a representation using all 500 of the FPS-SOAP components (marked in the graph as “Original”). In the case of predicting the energy per mol Si, upwards of 100 principal components are required to match the prediction accuracy of the representation containing all 500 FPS-SOAP vector elements. The convergence of the prediction accuracy to that of the full FPS-SOAP vector as more information (more principal components) are included into the KPCA representation also serves as a validation of the method: the KPCA-based representation can emulate the diversity of the SOAP vector and thus the local chemical environment.

A comparison can also be made between the prediction accuracy of the SOAP-KPCA representation and that of a classical descriptor with comparable information content (dimensionality). In this paradigm, a four-component SOAP-KPCA representation would contain

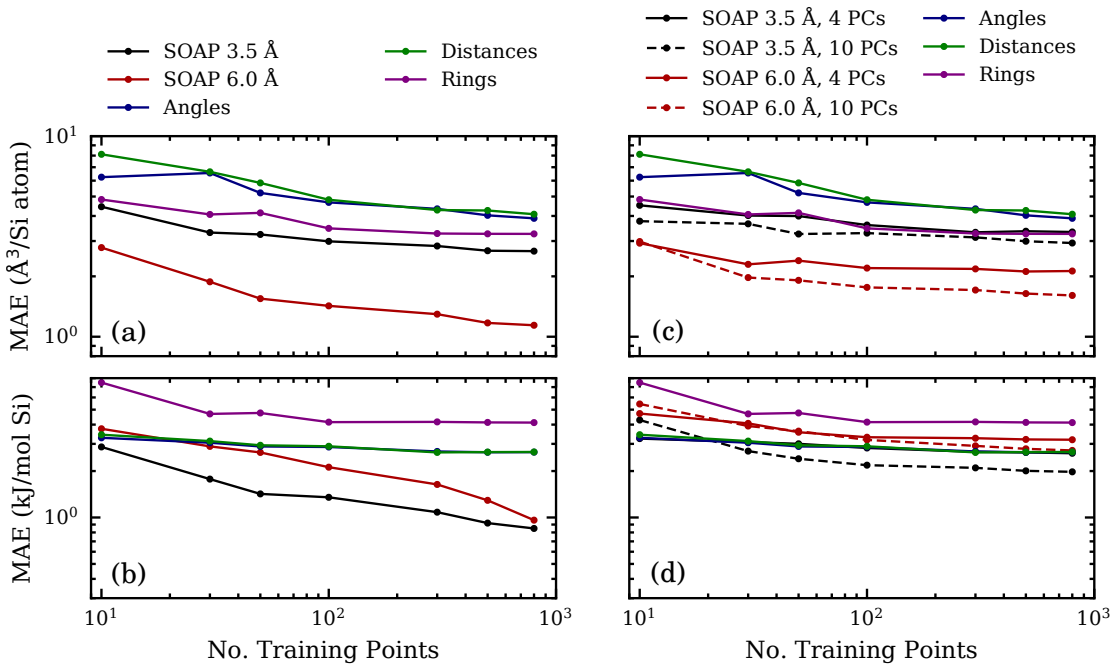

Figure C.2 – Learning curves of the classical and SOAP descriptors for predictions of (a) volume per Si atom and (b) energy per mol Si in the 1,000-structure subset. The error for each point in the learning curve calculated as the average of a five-fold cross-validation procedure using the optimal regularization and Gaussian kernel width. (c) and (d) re-plot the learning curves of the classical descriptors alongside the SOAP-KPCA descriptors with similar dimensionality (i.e., the number of features composing the representation). Adapted from Ref. [204] with permission of AIP publishing.

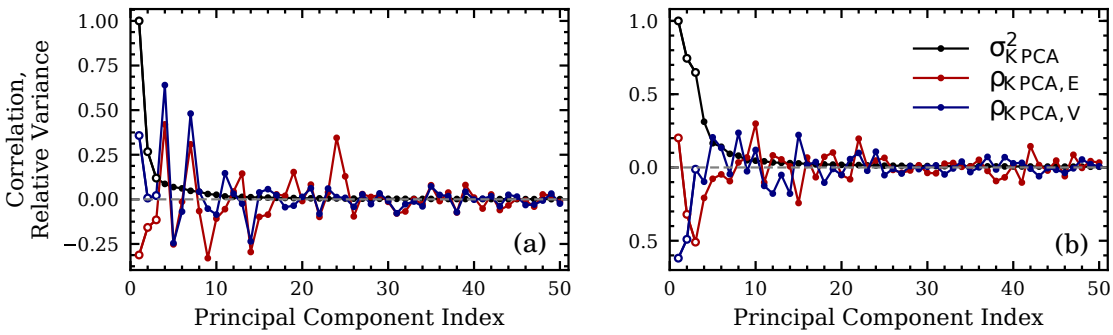

Figure C.3 – Pearson correlation coefficients between the first 50 KPCs of the (a) 3.5 Å SOAP representation and (b) 6.0 Å SOAP representation and the decomposed environment volumes and energies in the 1,000-structure sample. The relative variance in the KPCs at each of the first 50 components is also plotted. The correlation coefficients and relative variance of the first three components are highlighted with open symbols. Adapted from Ref. [204] with permission of AIP publishing.

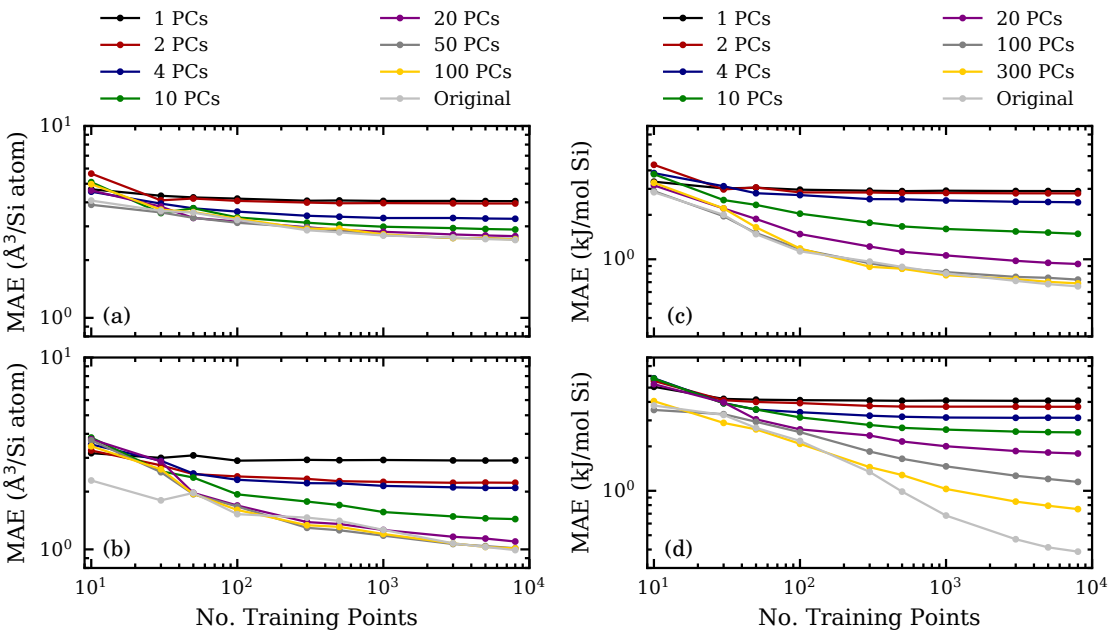

Figure C.4 – Learning curves for the SOAP-KPCA descriptors of various dimensionalities for (a)–(b) predicting the volume per Si and (c)–(d) the energy per mol Si in the 10,000-structure sample. The curves in (a) and (c) are based on SOAP descriptors with a cutoff radius of 3.5 Å, while those in (b) and (d) are based on SOAP descriptors with a cutoff radius of 6.0 Å. Increasing the amount of information embedded into the descriptor (increasing the number of principal components) results in better property predictions. Adapted from Ref. [204] with permission of AIP publishing.

roughly the same amount of information as the Si – O distance and Si – O – Si angle descriptors; a ten-component SOAP-KPCA representation would contain approximately the same amount of information as the ring-based descriptor, as ring sizes in our dataset range from 3 to 12. When comparing the different representations in this manner, one finds that the performance of the classical descriptors is comparable to, or slightly worse than, the performance of a 3.5 Å SOAP-KPCA descriptor including less than five principal components. The same is true for a comparison of volume predictions with the 6.0 Å SOAP-KPCA descriptor. In some cases, a SOAP-KPCA descriptor comprising only a single principal component outperforms one or more of the classical descriptors in predicting framework volumes or energies.

C.4 Synthesis

Fig. C.6 shows a histogram of the computed energies of the IZA frameworks from GULP [219], using the procedure described in 7.3.1. The energy of the framework RWY is much higher than any of the other frameworks on account of its reference composition containing neither Si nor O atoms. Considering that our GULP calculations use the SLC forcefield, tailored for Si – O interactions, the high reported energy is perhaps due in part to this compositional mismatch. Consequently, we discard RWY from our analyses involving IZA frameworks.

Fig. C.7 shows a histogram of the differences between our calculated energies and

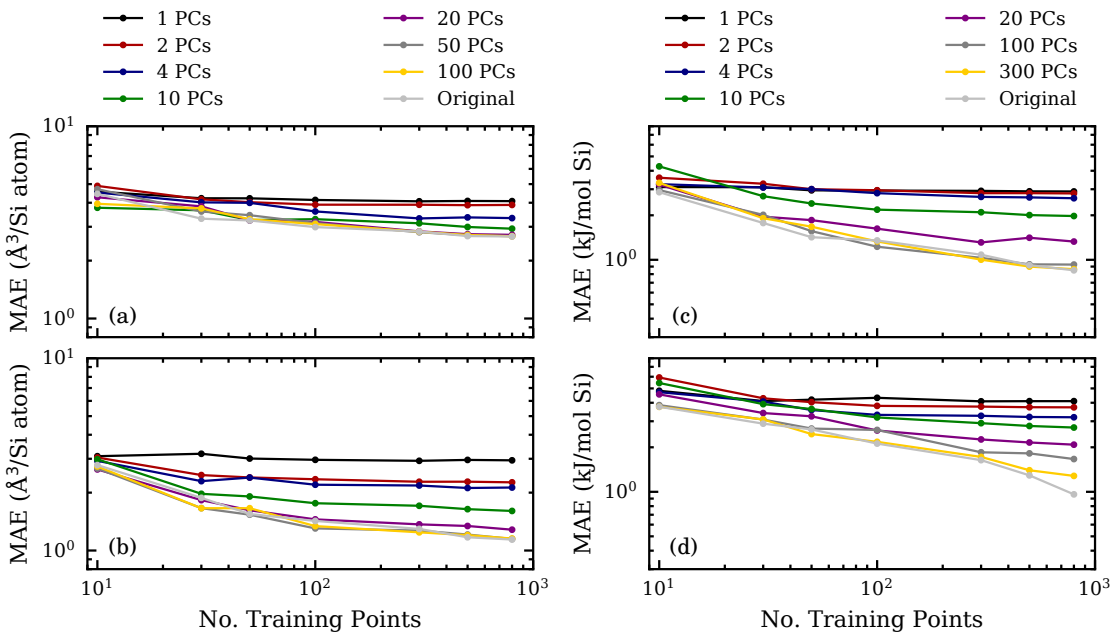

Figure C.5 – Learning curves for the SOAP-KPCA descriptors of various dimensionalities for (a)–(b) predicting the volume per Si and (c)–(d) the energy per mol Si in the 1,000-structure sample. The curves in (a) and (c) are based on SOAP descriptors with a cutoff radius of 3.5 Å, while those in (b) and (d) are based on SOAP descriptors with a cutoff radius of 6.0 Å. Increasing the amount of information embedded into the descriptor (increasing the number of principal components) results in better property predictions. Adapted from Ref. [204] with permission of AIP publishing.

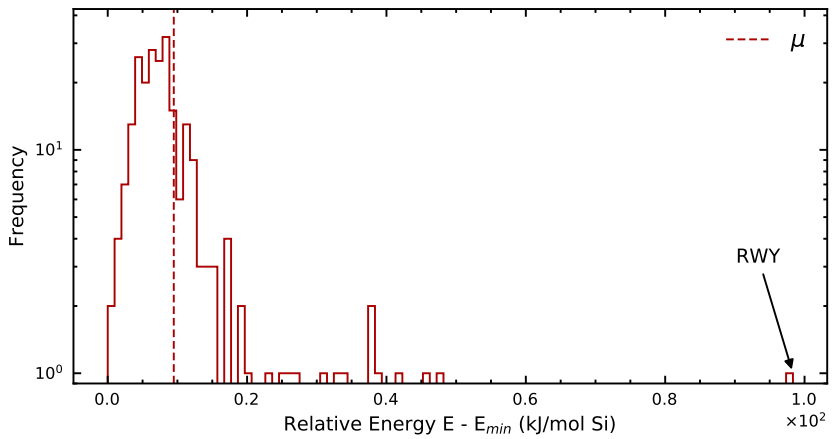

Figure C.6 – Histogram of IZA energies as computed with GULP. The computed energy for the framework RWY is considerably higher than all of the other frameworks.

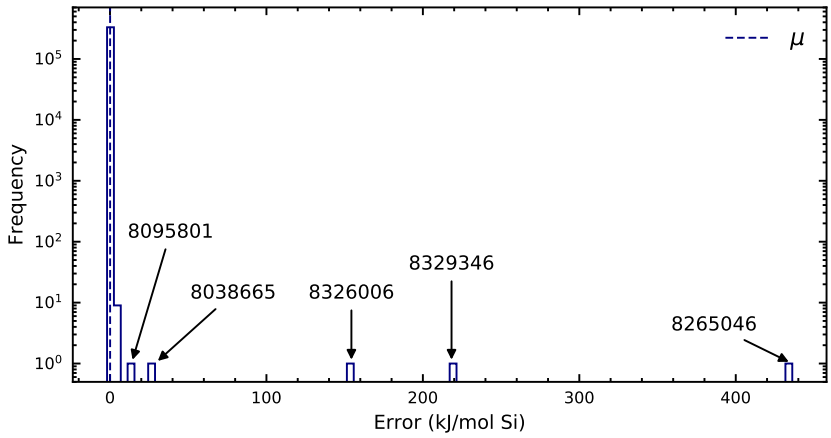

Figure C.7 – Histogram of errors representing the discrepancy between our GULP calculations of the framework molar energy for the approximately 330,000 structures in the Deem database of hypothetical zeolites. Structures with energy discrepancies larger than 10 kJ/mol Si are highlighted with their ID number.

the provided reference values for the frameworks in the Deem database. We are generally able to reproduce the database energies, but there are a few structures for which the energy discrepancy is quite high. We discard these structures, which have energy errors of more than 10 kJ/mol Si, from our subsequent analyses.

Fig. C.8 provides a histogram of the Euclidean distances between the IZA and Deem frameworks based on the 6.0 Å full power spectrum SOAP representation. We use this histogram to determine a cutoff for determining whether a given IZA framework exists in the Deem database. As the distribution of distances shows two peaks, with one at very small distances and the other at much larger distances, we select the cutoff for declaring identical structures to fall between the two peaks. We believe 5×10^{-6} to be a reasonable choice, so that a Deem framework at or closer than this distance to an IZA framework is considered identical to the IZA framework and is removed from subsequent analyses in order to avoid providing contradictory inputs to the classification models.

Fig. C.9 shows the histogram of decision functions, ROC curve, and confusion matrix for the two-class "IZA vs. Deem" support vector classification based on the SOAP power spectrum representation using an environment cutoff of 3.5 Å, similar to Figs. 7.9(a) and (b). The results of the classification on the 3.5 Å representation are generally similar to those of the 6.0 Å representation.

The confusion matrices of the four-class "IZA1 vs. IZA2 vs. IZA3 vs. Deem" cantonal classification for the ensemble of SVM models is shown in Fig. C.10, serving as an extension of the two-class case in Fig. 7.12 in Section 7.3.3. The SVM models have some difficulty accurately distinguishing between the different IZA "cantons", but still perform substantially better than a random guess.

To verify that the SVM classifications of IZA and Deem structures are based on genuine differences between the frameworks, we compute confusion matrices for a set of "dummy"

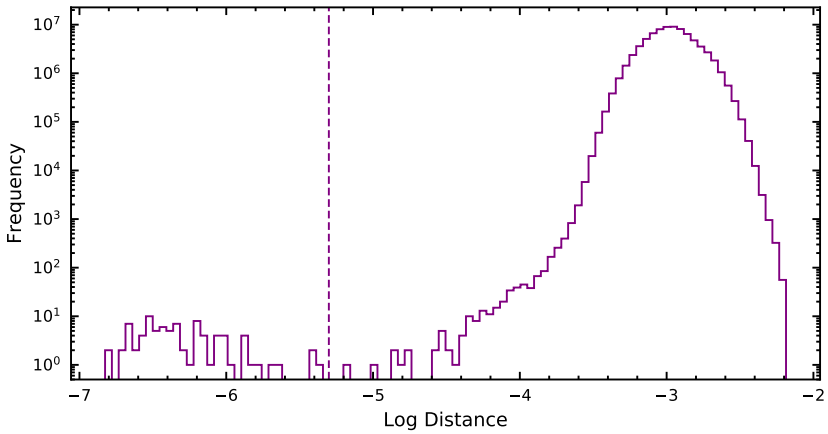

Figure C.8 – Histogram of Euclidean distances between the frameworks in the Deem database of hypothetical zeolites and the IZA structures. The distance is computed using the full power spectrum SOAP vectors of the 6.0 Å representation. The distance cutoff for declaring structures as "identical" is 5×10^{-6}.

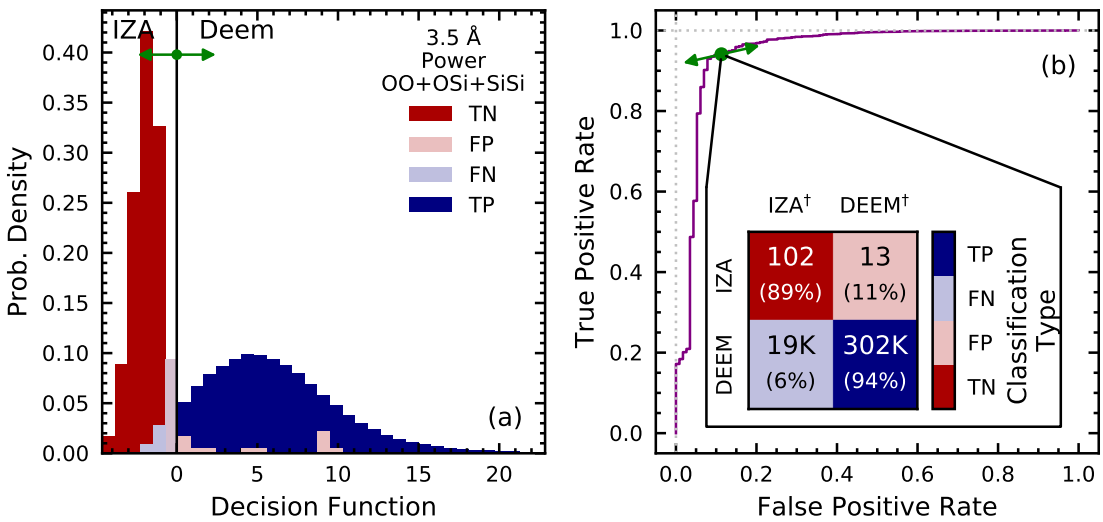

Figure C.9 – (a) Histogram of decision function values for IZA and Deem frameworks; (b) ROC curve for the "IZA vs. Deem" SVM classification based on a 3.5 Å SOAP representation as the decision function boundary is swept through the SOAP space. The inset of (b) also shows a confusion matrix for the two-class "IZA vs. Deem" classification using the full power spectrum SOAP vectors. The superscript † indicates predicted class labels.

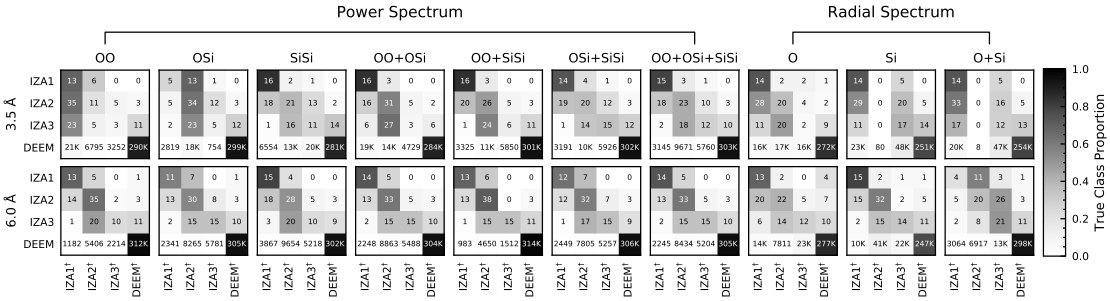

Figure C.10 – Confusion matrices from the four-class SVM classification where subsets of the power spectrum or radial spectrum were used for the SVM training and classification. The matrix entries are colored according to the proportion of the true labels that have been predicted as a particular class, while the interior text gives the absolute number of such classifications. While the classifier often correctly classifies the Deem frameworks as such, it has more difficulty distinguishing between the IZA subcategories. The superscript † indicates predicted class labels.

models, shown in Figs. C.11 and C.12. The dummy models are trained on the Deem frameworks in our train set (described in 7.3.3), but the target two- and four-class labels are assigned randomly. As evident from Figs. C.11 and C.12, in neither the two-class nor the four-class case is the SVM able to learn the random class distinctions, indicating that in the "IZA vs. Deem" confusion matrices of Figs. 7.12 and C.10 there are genuine distinctions between the IZA and Deem frameworks beyond random noise that the SVM picks up on in order to differentiate the classes.

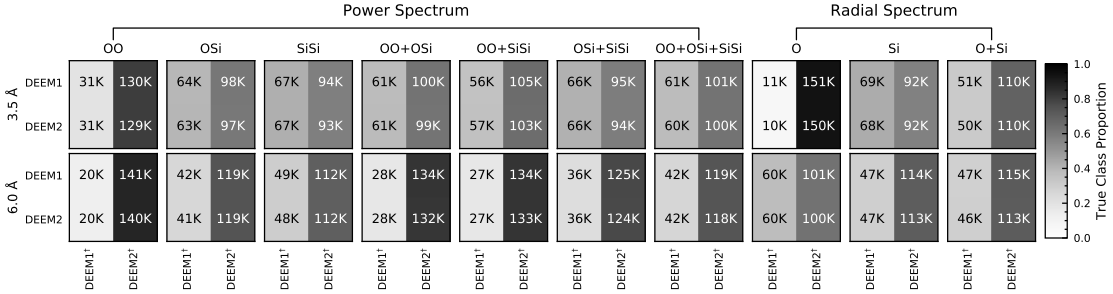

Figure C.11 – Confusion matrices from a two-class SVM classification where subsets of the power spectrum or radial spectrum were used for the SVM training and classification. The matrix entries are colored according to the proportion of the true labels that have been predicted as a particular class, while the interior text gives the absolute number of such classifications. The classification is trained on a subset of Deem frameworks that are assigned random class labels. The superscript † indicates predicted class labels.

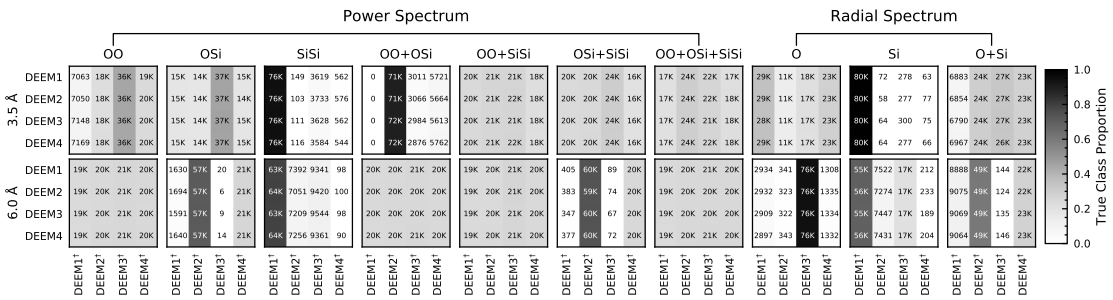

Figure C.12 – Confusion matrices from a four-class SVM classification where subsets of the power spectrum or radial spectrum were used for the SVM training and classification. The matrix entries are colored according to the proportion of the true labels that have been predicted as a particular class, while the interior text gives the absolute number of such classifications. The classification is trained on a subset of Deem frameworks that are assigned random class labels. The superscript † indicates predicted class labels.

Table C.1 – List of the 50 Deem frameworks that are closest to the convex hull that have a two-class decision function value F less than zero. For each candidate, its Deem database ID is given, in addition to its (energy) distance from the hull E_{hull}, two-class decision function value F, predicted four-class canton, and the closest IZA framework in SOAP space alongside its ground-truth assigned canton and its distance from the corresponding Deem framework D. The candidates are sorted by their two-class decision functions.

No.	ID	E_{hull}	F	Canton	Closest IZA (Canton)	D
(1)	8283748	0.00	-4.57	IZA2	VET (IZA1)	1.96×10^{-3}
(2)	8162069	0.00	-3.90	IZA3	VET (IZA1)	2.09×10^{-3}
(3)	8054476	0.00	-3.55	IZA2	SBN (IZA3)	2.41×10^{-4}
(4)	8214845	0.00	-3.20	IZA2	NAT (IZA2)	7.01×10^{-5}
(5)	8330882	0.00	-2.94	IZA1	MEP (IZA1)	4.58×10^{-4}
(6)	8330992	0.00	-2.85	IZA2	STF (IZA1)	6.56×10^{-4}
(7)	8315377	6.18×10^{-2}	-2.58	IZA3	AWW (IZA3)	3.99×10^{-4}
(8)	8315376	0.00	-2.53	IZA3	AWW (IZA3)	3.84×10^{-4}
(9)	8261336	1.22×10^{-1}	-2.46	IZA2	THO (IZA2)	2.33×10^{-4}
(10)	8122541	2.20×10^{-1}	-2.38	IZA2	POR (IZA3)	3.22×10^{-4}
(11)	8158735	0.00	-2.14	IZA2	THO (IZA2)	4.40×10^{-4}
(12)	8252698	3.26×10^{-1}	-1.70	IZA3	AWW (IZA3)	4.36×10^{-4}
(13)	8095665	1.74×10^{-1}	-1.59	IZA2	PTT (IZA2)	1.50×10^{-4}
(14)	8324141	1.57×10^{-1}	-1.50	IZA2	PTT (IZA2)	2.49×10^{-4}
(15)	8321610	1.18×10^{-1}	-1.44	IZA2	PTT (IZA2)	3.18×10^{-4}
(16)	8318169	3.17×10^{-1}	-1.40	IZA2	ATT (IZA3)	3.86×10^{-4}
(17)	8323694	0.00	-1.35	IZA2	FRA (IZA2)	2.58×10^{-4}
(18)	8227811	7.87×10^{-2}	-1.32	IZA1	IHW (IZA1)	4.88×10^{-4}
(19)	8322800	1.88×10^{-1}	-1.28	IZA3	FRA (IZA2)	3.83×10^{-4}
(20)	8327194	2.64×10^{-1}	-1.25	IZA2	FRA (IZA2)	7.85×10^{-5}
(21)	8322701	0.00	-1.25	IZA3	FRA (IZA2)	3.32×10^{-4}
(22)	8320027	0.00	-1.25	IZA2	PHI (IZA2)	6.67×10^{-5}
(23)	8322704	2.73×10^{-1}	-1.22	IZA3	GIU (IZA2)	3.90×10^{-4}
(24)	8170208	3.13×10^{-1}	-1.20	IZA3	AWO (IZA3)	4.19×10^{-4}
(25)	8327193	0.00	-1.19	IZA2	FRA (IZA2)	4.89×10^{-5}
(26)	8323749	1.49×10^{-1}	-1.13	IZA2	FRA (IZA2)	2.86×10^{-5}
(27)	8156062	0.00	-1.08	IZA3	PON (IZA3)	4.74×10^{-4}
(28)	8129131	2.48×10^{-1}	-1.07	IZA1	EWO (IZA2)	4.49×10^{-4}
(29)	8186781	2.37×10^{-1}	-1.00	IZA2	SEW (IZA2)	3.93×10^{-4}
(30)	8116170	2.08×10^{-2}	-0.98	IZA2	SIV (IZA3)	3.26×10^{-5}
(31)	8068062	3.49×10^{-2}	-0.98	IZA2	PHI (IZA2)	7.46×10^{-5}
(32)	8116169	1.85×10^{-2}	-0.96	IZA2	SIV (IZA3)	6.87×10^{-6}
(33)	8306691	0.00	-0.95	IZA2	MWF (IZA2)	1.36×10^{-4}
(34)	8238942	0.00	-0.92	IZA1	EWO (IZA2)	4.69×10^{-4}

(35)	8049770	2.97×10^{-1}	-0.90	IZA3	AWO (IZA3)	4.11×10^{-4}
(36)	8119960	1.99×10^{-3}	-0.88	IZA3	AWO (IZA3)	4.69×10^{-4}
(37)	8233794	0.00	-0.79	IZA3	AWO (IZA3)	4.69×10^{-4}
(38)	8169309	1.84×10^{-1}	-0.64	IZA1	EWO (IZA2)	4.70×10^{-4}
(39)	8011377	1.50×10^{-1}	-0.52	IZA3	MSO (IZA2)	5.32×10^{-4}
(40)	8073591	2.05×10^{-1}	-0.48	IZA3	AWO (IZA3)	3.01×10^{-4}
(41)	8168455	1.63×10^{-1}	-0.46	IZA3	PSI (IZA3)	6.43×10^{-4}
(42)	8097252	2.80×10^{-1}	-0.46	IZA3	AWO (IZA3)	3.03×10^{-4}
(43)	8192981	3.20×10^{-1}	-0.44	IZA3	SFG (IZA2)	6.29×10^{-4}
(44)	8243388	2.93×10^{-1}	-0.43	IZA3	AWO (IZA3)	2.08×10^{-4}
(45)	8076933	2.69×10^{-1}	-0.42	IZA2	GIS (IZA2)	3.20×10^{-4}
(46)	8118604	2.50×10^{-1}	-0.37	IZA3	AWO (IZA3)	3.07×10^{-4}
(47)	8129304	1.65×10^{-1}	-0.36	IZA1	EWO (IZA2)	4.99×10^{-4}
(48)	8050438	2.92×10^{-1}	-0.26	IZA3	AWO (IZA3)	5.08×10^{-4}
(49)	8125875	2.97×10^{-1}	-0.16	IZA3	MSO (IZA2)	4.81×10^{-4}
(50)	8073492	1.64×10^{-1}	-0.13	IZA3	AWO (IZA3)	2.72×10^{-4}

D Computational Tools

The data analysis and and visualization for the work presented in this thesis was performed in Python [240, 241] with the aid of the SciPy [242], NumPy [243–245], scikit-learn [202], Atomic Simulation Environment (ASE) [246], Biopython [199], and Matplotlib [247] packages in addition to Wolfram Mathematica 11.1 [248]. Atomic structure snapshots were created with OVITO [249], VESTA [250], or Visual Molecular Dynamics (VMD) [251] with the Tachyon [252] rendering utility.

References

1. Magee, C. L. The Role of Materials Innovation in Overall Technological Development. *JOM* **62,** 20–24. doi:10.1007/s11837-010-0043-5 (2010).
2. Maine, E. & Garnsey, E. Commercializing Generic Technology: The Case of Advanced Materials Ventures. *Research Policy* **35,** 375–393. doi:10.1016/j.respol.2005.12.006 (2006).
3. Tabor, D. P. *et al.* Accelerating the Discovery of Materials for Clean Energy in the Era of Smart Automation. *Nature Reviews Materials* **3,** 5–20. doi:10.1038/s41578-018-0005-z (2018).
4. Rupp, M., Tkatchenko, A., Müller, K.-R. & von Lilienfeld, O. A. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Physical Review Letters* **108,** 058301. doi:10.1103/PhysRevLett.108.058301 (2012).
5. Hansen, K. *et al.* Assessment and Validation of Machine Learning Methods for Predicting Molecular Atomization Energies. *Journal of Chemical Theory and Computation* **9,** 3404–3419. doi:10.1021/ct400195d (2013).
6. Hansen, K. *et al.* Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space. *The Journal of Physical Chemistry Letters* **6,** 2326–2331. doi:10.1021/acs.jpclett.5b00831 (2015).
7. Faber, F. A., Lindmaa, A., von Lilienfeld, O. A. & Armiento, R. Machine Learning Energies of 2 Million Elpasolite (A B C 2 D 6) Crystals. *Physical Review Letters* **117,** 135502. doi:10.1103/PhysRevLett.117.135502 (2016).
8. Faber, F. A. *et al.* Prediction Errors of Molecular Machine Learning Models Lower than Hybrid DFT Error. *Journal of Chemical Theory and Computation* **13,** 5255–5264. doi:10.1021/acs.jctc.7b00577 (2017).
9. Faber, F. A., Christensen, A. S., Huang, B. & von Lilienfeld, O. A. Alchemical and Structural Distribution Based Representation for Universal Quantum Machine Learning. *The Journal of Chemical Physics* **148,** 241717. doi:10.1063/1.5020710 (2018).
10. Huan, T. D., Mannodi-Kanakkithodi, A. & Ramprasad, R. Accelerated Materials Property Predictions and Design Using Motif-Based Fingerprints. *Physical Review B* **92,** 014106. doi:10.1103/PhysRevB.92.014106 (2015).
11. Isayev, O. *et al.* Universal Fragment Descriptors for Predicting Properties of Inorganic Crystals. *Nature Communications* **8,** 15679. doi:10.1038/ncomms15679 (2017).

12. Von Lilienfeld, O. A., Ramakrishnan, R., Rupp, M. & Knoll, A. Fourier Series of Atomic Radial Distribution Functions: A Molecular Fingerprint for Machine Learning Models of Quantum Chemical Properties. *International Journal of Quantum Chemistry* **115,** 1084–1093. doi:10.1002/qua.24912 (2015).
13. Xie, T. & Grossman, J. C. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Physical Review Letters* **120,** 145301. doi:10.1103/PhysRevLett.120.145301 (2018).
14. Bartók, A. P. *et al.* Machine Learning Unifies the Modeling of Materials and Molecules. *Science Advances* **3,** e1701816. doi:10.1126/sciadv.1701816 (2017).
15. Freitas, R. & Reed, E. J. Uncovering the Effects of Interface-Induced Ordering of Liquid on Crystal Growth Using Machine Learning. *Nature Communications* **11,** 3260. doi:10.1038/s41467-020-16892-4 (2020).
16. Evans, J. D. & Coudert, F.-X. Predicting the Mechanical Properties of Zeolite Frameworks by Machine Learning. *Chemistry of Materials* **29,** 7833–7839. doi:10.1021/acs.chemmater.7b02532 (2017).
17. Paruzzo, F. M. *et al.* Chemical Shifts in Molecular Solids by Machine Learning. *Nature Communications* **9,** 4501. doi:10.1038/s41467-018-06972-x (2018).
18. Gao, P., Zhang, J., Peng, Q., Zhang, J. & Glezakou, V.-A. General Protocol for the Accurate Prediction of Molecular 13C/1H NMR Chemical Shifts via Machine Learning Augmented DFT. *Journal of Chemical Information and Modeling* **60,** 3746–3754. doi:10.1021/acs.jcim.0c00388 (2020).
19. Grisafi, A., Wilkins, D. M., Csányi, G. & Ceriotti, M. Symmetry-Adapted Machine Learning for Tensorial Properties of Atomistic Systems. *Physical Review Letters* **120,** 036002. doi:10.1103/PhysRevLett.120.036002 (2018).
20. Veit, M., Wilkins, D. M., Yang, Y., DiStasio, R. A. & Ceriotti, M. Predicting Molecular Dipole Moments by Combining Atomic Partial Charges and Atomic Dipoles. *The Journal of Chemical Physics* **153,** 024113. doi:10.1063/5.0009106 (2020).
21. Wilkins, D. M. *et al.* Accurate Molecular Polarizabilities with Coupled Cluster Theory and Machine Learning. *Proceedings of the National Academy of Sciences* **116,** 3401–3406. doi:10.1073/pnas.1816132116. pmid: 30733292 (2019).
22. Grisafi, A. *et al.* Transferable Machine-Learning Model of the Electron Density. *ACS Central Science* **5,** 57–64. doi:10.1021/acscentsci.8b00551 (2019).
23. Tsubaki, M. & Mizoguchi, T. Quantum Deep Field: Data-Driven Wave Function, Electron Density Generation, and Atomization Energy Prediction and Extrapolation with Machine Learning. *Physical Review Letters* **125,** 206401. doi:10.1103/PhysRevLett.125.206401 (2020).
24. Brockherde, F. *et al.* Bypassing the Kohn-Sham Equations with Machine Learning. *Nature Communications* **8,** 872. doi:10.1038/s41467-017-00839-3 (2017).

25. Chandrasekaran, A. *et al.* Solving the Electronic Structure Problem with Machine Learning. *npj Computational Materials* **5,** 1–7. doi:10.1038/s41524-019-0162-7 (2019).

26. Ben Mahmoud, C., Anelli, A., Csányi, G. & Ceriotti, M. Learning the Electronic Density of States in Condensed Matter. *Physical Review B* **102,** 235130. doi:10.1103/PhysRevB.102.235130 (2020).

27. Schütt, K. T. *et al.* How to Represent Crystal Structures for Machine Learning: Towards Fast Prediction of Electronic Properties. *Physical Review B* **89,** 205118. doi:10.1103/PhysRevB.89.205118 (2014).

28. Schütt, K. T., Gastegger, M., Tkatchenko, A., Müller, K.-R. & Maurer, R. J. Unifying Machine Learning and Quantum Chemistry with a Deep Neural Network for Molecular Wavefunctions. *Nature Communications* **10,** 5024. doi:10.1038/s41467-019-12875-2 (2019).

29. Choo, K., Mezzacapo, A. & Carleo, G. Fermionic Neural-Network States for Ab-Initio Electronic Structure. *Nature Communications* **11,** 2368. doi:10.1038/s41467-020-15724-9 (2020).

30. Hermann, J., Schätzle, Z. & Noé, F. Deep-Neural-Network Solution of the Electronic Schrödinger Equation. *Nature Chemistry* **12,** 891–897. doi:10.1038/s41557-020-0544-y (2020).

31. Pfau, D., Spencer, J. S., Matthews, A. G. D. G. & Foulkes, W. M. C. Ab Initio Solution of the Many-Electron Schrödinger Equation with Deep Neural Networks. *Physical Review Research* **2,** 033429. doi:10.1103/PhysRevResearch.2.033429 (2020).

32. Bartók, A. P., Kermode, J., Bernstein, N. & Csányi, G. Machine Learning a General-Purpose Interatomic Potential for Silicon. *Physical Review X* **8,** 041048. doi:10.1103/PhysRevX.8.041048 (2018).

33. Deringer, V. L. & Csányi, G. Machine Learning Based Interatomic Potential for Amorphous Carbon. *Physical Review B* **95,** 094203. doi:10.1103/PhysRevB.95.094203 (2017).

34. Dragoni, D., Daff, T. D., Csányi, G. & Marzari, N. Achieving DFT Accuracy with a Machine-Learning Interatomic Potential: Thermomechanics and Defects in Bcc Ferromagnetic Iron. *Physical Review Materials* **2,** 013808. doi:10.1103/PhysRevMaterials.2.013808 (2018).

35. Maillet, J.-B., Denoual, C. & Csányi, G. Machine-Learning Based Potential for Iron: Plasticity and Phase Transition. *AIP Conference Proceedings* **1979,** 050011. doi:10.1063/1.5044794 (2018).

36. Szlachta, W. J., Bartók, A. P. & Csányi, G. Accuracy and Transferability of Gaussian Approximation Potential Models for Tungsten. *Physical Review B* **90,** 104108. doi:10.1103/PhysRevB.90.104108 (2014).

37. Schütt, O. & VandeVondele, J. Machine Learning Adaptive Basis Sets for Efficient Large Scale Density Functional Theory Simulation. *Journal of Chemical Theory and Computation* **14,** 4168–4175. doi:10.1021/acs.jctc.8b00378 (2018).

38. Botu, V. & Ramprasad, R. Adaptive Machine Learning Framework to Accelerate *Ab Initio* Molecular Dynamics. *International Journal of Quantum Chemistry* **115,** 1074–1083. doi:10.1002/qua.24836 (2015).

39. He, T. *et al.* Similarity of Precursors in Solid-State Synthesis as Text-Mined from Scientific Literature. *Chemistry of Materials* **32,** 7861–7873. doi:10.1021/acs.chemmater.0c02553 (2020).

40. Kim, E. *et al.* Materials Synthesis Insights from Scientific Literature via Text Extraction and Machine Learning. *Chemistry of Materials* **29,** 9436–9444. doi:10.1021/acs.chemmater.7b03500 (2017).

41. Kononova, O. *et al.* Text-Mined Dataset of Inorganic Materials Synthesis Recipes. *Scientific Data* **6,** 203. doi:10.1038/s41597-019-0224-1 (2019).

42. Kim, E., Huang, K., Jegelka, S. & Olivetti, E. Virtual Screening of Inorganic Materials Synthesis Parameters with Deep Learning. *npj Computational Materials* **3,** 1–9. doi:10.1038/s41524-017-0055-6 (2017).

43. Kim, E. *et al.* Machine-Learned and Codified Synthesis Parameters of Oxide Materials. *Scientific Data* **4,** 170127. doi:10.1038/sdata.2017.127 (2017).

44. Kim, E. *et al.* Inorganic Materials Synthesis Planning with Literature-Trained Neural Networks. *Journal of Chemical Information and Modeling* **60,** 1194–1201. doi:10.1021/acs.jcim.9b00995 (2020).

45. Huo, H. *et al.* Semi-Supervised Machine-Learning Classification of Materials Synthesis Procedures. *npj Computational Materials* **5,** 1–7. doi:10.1038/s41524-019-0204-1 (2019).

46. Jensen, Z. *et al.* A Machine Learning Approach to Zeolite Synthesis Enabled by Automatic Literature Data Extraction. *ACS Central Science* **5,** 892–899. doi:10.1021/acscentsci.9b00193 (2019).

47. Sendek, A. D. *et al.* Holistic Computational Structure Screening of More than 12 000 Candidates for Solid Lithium-Ion Conductor Materials. *Energy & Environmental Science* **10,** 306–320. doi:10.1039/C6EE02697D (2017).

48. Fernandez, M., Boyd, P. G., Daff, T. D., Aghaji, M. Z. & Woo, T. K. Rapid and Accurate Machine Learning Recognition of High Performing Metal Organic Frameworks for CO2 Capture. *The Journal of Physical Chemistry Letters* **5,** 3056–3060. doi:10.1021/jz501331m (2014).

49. Tshitoyan, V. *et al.* Unsupervised Word Embeddings Capture Latent Knowledge from Materials Science Literature. *Nature* **571,** 95–98. doi:10.1038/s41586-019-1335-8 (2019).

50. Ruddigkeit, L., van Deursen, R., Blum, L. C. & Reymond, J.-L. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *Journal of Chemical Information and Modeling* **52,** 2864–2875. doi:10.1021/ci300415d (2012).

51. Ramakrishnan, R., Dral, P. O., Rupp, M. & von Lilienfeld, O. A. Quantum Chemistry Structures and Properties of 134 Kilo Molecules. *Scientific Data* **1,** 140022. doi:10.1038/sdata.2014.22 (2014).

52. Ramakrishnan, R., Dral, P., Rupp, M. & von Lilienfeld, O. A. Quantum Chemistry Structures and Properties of 134 Kilo Molecules. doi:10.6084/M9.FIGSHARE.C.978904.V5 (2019).

53. Groom, C. R., Bruno, I. J., Lightfoot, M. P. & Ward, S. C. The Cambridge Structural Database. *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials* **72,** 171–179. doi:10.1107/S2052520616003954 (2016).

54. Deem, M. W., Pophale, R., Cheeseman, P. A. & Earl, D. J. Computational Discovery of New Zeolite-Like Materials. *The Journal of Physical Chemistry C* **113,** 21353–21360. doi:10.1021/jp906984z (2009).

55. Earl, D. J. & Deem, M. W. Toward a Database of Hypothetical Zeolite Structures. *Industrial & Engineering Chemistry Research* **45,** 5449–5454. doi:10.1021/ie0510728 (2006).

56. Pophale, R., Cheeseman, P. A. & Deem, M. W. A Database of New Zeolite-like Materials. *Physical Chemistry Chemical Physics* **13,** 12407–12412. doi:10.1039/C0CP02255A (2011).

57. Foster, M. D. & Treacy, M. M. J. in *Studies in Surface Science and Catalysis* (eds Xu, R., Gao, Z., Chen, J. & Yan, W.) 666–673 (Elsevier, 2007). doi:10.1016/S0167-2991(07)80906-2.

58. Treacy, M. M. J., Rivin, I., Balkovsky, E., Randall, K. H. & Foster, M. D. Enumeration of Periodic Tetrahedral Frameworks. II. Polynodal Graphs. *Microporous and Mesoporous Materials* **74,** 121–132. doi:10.1016/j.micromeso.2004.06.013 (2004).

59. Treacy, M. M. J., Randall, K. H., Rao, S., Perry, J. A. & Chadi, D. J. Enumeration of Periodic Tetrahedral Frameworks. *Zeitschrift für Kristallographie - Crystalline Materials* **212,** 768–791. doi:10.1524/zkri.1997.212.11.768 (1997).

60. Ropo, M., Schneider, M., Baldauf, C. & Blum, V. First-Principles Data Set of 45,892 Isolated and Cation-Coordinated Conformers of 20 Proteinogenic Amino Acids. *Scientific Data* **3,** 160009. doi:10.1038/sdata.2016.9 (2016).

61. Baldauf, C. NoMaD Repository Entry. doi:10.17172/NOMAD/20150526220502 (2015).

62. Pickard, C. J. & Needs, R. J. Ab Initio Random Structure Searching. *Journal of Physics: Condensed Matter* **23,** 053201. doi:10.1088/0953-8984/23/5/053201 (2011).

63. Pickard, C. J. *AIRSS Data for Carbon at 10GPa and the C+N+H+O System at 1GPa* (Materials Cloud, 2020). doi:10.24435/MATERIALSCLOUD:2020.0026/V1.

64. Berman, H. M. *et al.* The Protein Data Bank. *Nucleic Acids Research* **28,** 235–242. doi:10.1093/nar/28.1.235 (2000).

65. Gražulis, S. *et al.* Crystallography Open Database – an Open-Access Collection of Crystal Structures. *Journal of Applied Crystallography* **42,** 726–729. doi:10.1107/S0021889809016690 (2009).

66. Gražulis, S. *et al.* Crystallography Open Database (COD): An Open-Access Collection of Crystal Structures and Platform for World-Wide Collaboration. *Nucleic Acids Research* **40,** D420–D427. doi:10.1093/nar/gkr900 (2012).

67. Gražulis, S., Merkys, A., Vaitkus, A. & Okulič-Kazarinas, M. Computing Stoichiometric Molecular Composition from Crystal Structures. *Journal of Applied Crystallography* **48,** 85–91. doi:10.1107/S1600576714025904 (2015).

68. Downs, R. T. & Hall-Wallace, M. The American Mineralogist Crystal Structure Database. *American Mineralogist* **88,** 247–250 (2003).

69. Merkys, A. *et al.* COD::CIF::Parser: An Error-Correcting CIF Parser for the Perl Language. *Journal of Applied Crystallography* **49,** 292–301. doi:10.1107/S1600576715022396 (2016).

70. Quirós, M., Gražulis, S., Girdzijauskaitė, S., Merkys, A. & Vaitkus, A. Using SMILES Strings for the Description of Chemical Connectivity in the Crystallography Open Database. *Journal of Cheminformatics* **10,** 23. doi:10.1186/s13321-018-0279-6 (2018).

71. Vaitkus, A., Merkys, A. & Gražulis, S. Validation of the Crystallography Open Database Using the Crystallographic Information Framework. *Journal of Applied Crystallography* **54.** doi:10.1107/S1600576720016532 (2021).

72. Murphy, K. P. *Machine Learning: A Probabilistic Perspective* (MIT Press, Cambridge, MA, 2012).

73. Bishop, C. M. *Pattern Recognition and Machine Learning* (Springer, New York, 2006).

74. Hastie, T., Tibshirani, R. & Friedman, J. H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* 2nd ed (Springer, New York, NY, 2009).

75. Chapelle, O., Schölkopf, B. & Zien, A. *Semi-Supervised Learning* (MIT Press, Cambridge, MA, 2010).

76. Ghiringhelli, L. M., Vybiral, J., Levchenko, S. V., Draxl, C. & Scheffler, M. Big Data of Materials Science: Critical Role of the Descriptor. *Physical Review Letters* **114,** 105503. doi:10.1103/PhysRevLett.114.105503 (2015).

77. Faber, F., Lindmaa, A., von Lilienfeld, O. A. & Armiento, R. Crystal Structure Representations for Machine Learning Models of Formation Energies. *International Journal of Quantum Chemistry* **115,** 1094–1101. doi:10.1002/qua.24917 (2015).

78. Huang, B. & von Lilienfeld, O. A. Communication: Understanding Molecular Representations in Machine Learning: The Role of Uniqueness and Target Similarity. *The Journal of Chemical Physics* **145,** 161102. doi:10.1063/1.4964627 (2016).

79. Ward, L. *et al.* Including Crystal Structure Attributes in Machine Learning Models of Formation Energies via Voronoi Tessellations. *Physical Review B* **96,** 024104. doi:10.1103/PhysRevB.96.024104 (2017).

80. Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *Journal of Chemical Information and Computer Sciences* **28,** 31–36. doi:10.1021/ci00057a005 (1988).

81. Gómez-Bombarelli, R. *et al.* Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science* **4,** 268–276. doi:10.1021/acscentsci.7b00572 (2018).

82. Huo, H. & Rupp, M. *Unified Representation of Molecules and Crystals for Machine Learning* arXiv: 1704.06439 [cond-mat, physics:physics].

83. Behler, J. & Parrinello, M. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Physical Review Letters* **98,** 146401. doi:10.1103/PhysRevLett.98.146401 (2007).

84. Bartók, A. P., Kondor, R. & Csányi, G. On Representing Chemical Environments. *Physical Review B* **87,** 184115. doi:10.1103/PhysRevB.87.184115 (2013).

85. De, S., Bartók, A. P., Csányi, G. & Ceriotti, M. Comparing Molecules and Solids across Structural and Alchemical Space. *Physical Chemistry Chemical Physics* **18,** 13754–13769. doi:10.1039/C6CP00415F (2016).

86. Willatt, M. J., Musil, F. & Ceriotti, M. Feature Optimization for Atomistic Machine Learning Yields a Data-Driven Construction of the Periodic Table of the Elements. *Physical Chemistry Chemical Physics* **20,** 29661–29668. doi:10.1039/C8CP05921G (2018).

87. Musil, F. & Ceriotti, M. Machine Learning at the Atomic Scale. *CHIMIA International Journal for Chemistry* **73,** 972–982. doi:10.2533/chimia.2019.972 (2019).

88. Willatt, M. J., Musil, F. & Ceriotti, M. Atom-Density Representations for Machine Learning. *The Journal of Chemical Physics* **150,** 154110. doi:10.1063/1.5090481 (2019).

89. Musil, F. *et al. Physics-Inspired Structural Representations for Molecules and Materials* arXiv: 2101.04673 [physics].

90. Musil, F. *et al.* Machine Learning for the Structure–Energy–Property Landscapes of Molecular Crystals. *Chemical Science* **9,** 1289–1300. doi:10.1039/C7SC04665K (2018).

91. Gasparotto, P., Bochicchio, D., Ceriotti, M. & Pavan, G. M. Identifying and Tracking Defects in Dynamic Supramolecular Polymers. *The Journal of Physical Chemistry B* **124,** 589–599. doi:10.1021/acs.jpcb.9b11015 (2020).

92. Rowe, P., Deringer, V. L., Gasparotto, P., Csányi, G. & Michaelides, A. An Accurate and Transferable Machine Learning Potential for Carbon. *The Journal of Chemical Physics* **153,** 034702. doi:10.1063/5.0005084 (2020).

93. Musil, F. *et al.* Efficient Implementation of Atom-Density Representations. *The Journal of Chemical Physics* **154,** 114109. doi:10.1063/5.0044689 (2021).

94. Goscinski, A., Musil, F., Pozdnyakov, S. & Ceriotti, M. *Optimal Radial Basis for Density-Based Atomic Representations* arXiv: 2105.08717 [physics, stat].

95. Schölkopf, B. & Smola, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* (MIT Press, Cambridge, Mass, 2002).

96. Mercer, J. & Forsyth, A. R. XVI. Functions of Positive and Negative Type, and Their Connection the Theory of Integral Equations. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* **209,** 415–446. doi:10.1098/rsta.1909.0016 (1909).

97. Aronszajn, N. Theory of Reproducing Kernels. *Transactions of the American Mathematical Society* **68,** 337–404. doi:10.1090/S0002-9947-1950-0051437-7 (1950).

98. Kimeldorf, G. S. & Wahba, G. A Correspondence Between Bayesian Estimation on Stochastic Processes and Smoothing by Splines. *The Annals of Mathematical Statistics* **41,** 495–502. doi:10.1214/aoms/1177697089 (1970).

99. Kimeldorf, G. & Wahba, G. Some Results on Tchebycheffian Spline Functions. *Journal of Mathematical Analysis and Applications* **33,** 82–95. doi:10.1016/0022-247X(71)90184-3 (1971).

100. Schölkopf, B., Herbrich, R. & Smola, A. J. *A Generalized Representer Theorem* in *Computational Learning Theory* (eds Helmbold, D. & Williamson, B.) (Springer, Berlin, Heidelberg, 2001), 416–426. doi:10.1007/3-540-44581-1_27.

101. Williams, C. K. I. & Seeger, M. in *Advances in Neural Information Processing Systems 13* (eds Leen, T. K., Dietterich, T. G. & Tresp, V.) 682–688 (MIT Press, 2001).

102. Tipping, M. E. in *Advances in Neural Information Processing Systems 13* (eds Leen, T. K., Dietterich, T. G. & Tresp, V.) 633–639 (MIT Press, 2001).

103. Drineas, P. & Mahoney, M. W. On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning. *Journal of Machine Learning Research* **6,** 2153–2175 (2005).

104. Smola, A. J. & Schökopf, B. *Sparse Greedy Matrix Approximation for Machine Learning* in *Proceedings of the Seventeenth International Conference on Machine Learning* (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000), 911–918.

105. Zhang, K., Tsang, I. W. & Kwok, J. T. *Improved Nyström Low-Rank Approximation and Error Analysis* in *Proceedings of the 25th International Conference on Machine Learning* Helsinki, Finland (ACM, New York, NY, USA, 2008), 1232–1239. doi:10.1145/1390156.1390311.

106. Eldar, Y., Lindenbaum, M., Porat, M. & Zeevi, Y. Y. The Farthest Point Strategy for Progressive Image Sampling. *IEEE Transactions on Image Processing* **6,** 1305–1315. doi:10.1109/83.623193 (1997).

107. Helfrecht, B. A., Cersonsky, R. K., Fraux, G. & Ceriotti, M. Structure-Property Maps with Kernel Principal Covariates Regression. *Machine Learning: Science and Technology* **1,** 045021. doi:10.1088/2632-2153/aba9ef (2020).

108. Ceriotti, M., Willatt, M. J. & Csányi, G. in *Handbook of Materials Modeling* (eds Andreoni, W. & Yip, S.) 1–27 (Springer International Publishing, Cham, 2018). doi:10.1007/978-3-319-42913-7_68-1.

109. Kohonen, T. Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics* **43,** 59–69. doi:10.1007/BF00337288 (1982).

110. Gasparotto, P. & Ceriotti, M. Recognizing Molecular Patterns by Machine Learning: An Agnostic Structural Definition of the Hydrogen Bond. *The Journal of Chemical Physics* **141,** 174110. doi:10.1063/1.4900655 (2014).

111. Gasparotto, P., Meißner, R. H. & Ceriotti, M. Recognizing Local and Global Structural Motifs at the Atomic Scale. *Journal of Chemical Theory and Computation* **14,** 486–498. doi:10.1021/acs.jctc.7b00993 (2018).

112. Chen, Y., Wiesel, A., Eldar, Y. C. & Hero, A. O. Shrinkage Algorithms for MMSE Covariance Estimation. *IEEE Transactions on Signal Processing* **58,** 5016–5029. doi:10.1109/TSP.2010.2053029 (2010).

113. Vedaldi, A. & Soatto, S. *Quick Shift and Kernel Methods for Mode Seeking* in *Computer Vision – ECCV 2008* (eds Forsyth, D., Torr, P. & Zisserman, A.) (Springer, Berlin, Heidelberg, 2008), 705–718. doi:10.1007/978-3-540-88693-8_52.

114. Roweis, S. T. & Saul, L. K. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* **290,** 2323–2326. doi:10.1126/science.290.5500.2323. pmid: 11125150 (2000).

115. Tenenbaum, J. B., de Silva, V. & Langford, J. C. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* **290,** 2319–2323. doi:10.1126/science.290.5500.2319. pmid: 11125149 (2000).

116. Hinton, G. E. & Roweis, S. Stochastic Neighbor Embedding. *Advances in Neural Information Processing Systems* **15** (2002).

117. Van der Maaten, L. & Hinton, G. Visualizing Data Using T-SNE. *Journal of Machine Learning Research* **9,** 2579–2605 (2008).

118. Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise* in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (AAAI Press, Portland, Oregon, 1996), 226–231.

119. Campello, R. J. G. B., Moulavi, D. & Sander, J. *Density-Based Clustering Based on Hierarchical Density Estimates* in *Advances in Knowledge Discovery and Data Mining* (eds Pei, J., Tseng, V. S., Cao, L., Motoda, H. & Xu, G.) (Springer, Berlin, Heidelberg, 2013), 160–172. doi:10.1007/978-3-642-37456-2_14.

120. Campello, R. J. G. B., Moulavi, D., Zimek, A. & Sander, J. Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. *ACM Transactions on Knowledge Discovery from Data* **10,** 5:1–5:51. doi:10.1145/2733381 (2015).

121. Ceriotti, M., Tribello, G. A. & Parrinello, M. Simplifying the Representation of Complex Free-Energy Landscapes Using Sketch-Map. *Proceedings of the National Academy of Sciences* **108,** 13023–13028. doi:10.1073/pnas.1108486108. pmid: 21730167 (2011).

122. Tribello, G. A., Ceriotti, M. & Parrinello, M. Using Sketch-Map Coordinates to Analyze and Bias Molecular Dynamics Simulations. *Proceedings of the National Academy of Sciences* **109,** 5196–5201. doi:10.1073/pnas.1201152109. pmid: 22427357 (2012).

123. Ceriotti, M., Tribello, G. A. & Parrinello, M. Demonstrating the Transferability and the Descriptive Power of Sketch-Map. *Journal of Chemical Theory and Computation* **9,** 1521–1532. doi:10.1021/ct3010563 (2013).

124. F.R.S, K. P. LIII. On Lines and Planes of Closest Fit to Systems of Points in Space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **2,** 559–572. doi:10.1080/14786440109462720 (1901).

125. Hotelling, H. Analysis of a Complex of Statistical Variables into Principal Components. *Journal of Educational Psychology* **24,** 417–441. doi:10.1037/h0071325 (1933).

126. Schölkopf, B., Smola, A. & Müller, K. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation* **10,** 1299–1319. doi:10.1162/089976698300017467 (1998).

127. Torgerson, W. S. Multidimensional Scaling: I. Theory and Method. *Psychometrika* **17,** 401–419. doi:10.1007/BF02288916 (1952).

128. Cortes, C. & Vapnik, V. Support-Vector Networks. *Machine Learning* **20,** 273–297. doi:10.1007/BF00994018 (1995).

129. Tikhonov, A. N., Goncharsky, A. V., Stepanov, V. V. & Yagola, A. G. in *Numerical Methods for the Solution of Ill-Posed Problems* (eds Tikhonov, A. N., Goncharsky, A. V., Stepanov, V. V. & Yagola, A. G.) 7–63 (Springer Netherlands, Dordrecht, 1995). doi:10.1007/978-94-015-8480-7_2.

130. Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* **58,** 267–288. JSTOR: 2346178 (1996).

131. Zou, H. & Hastie, T. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67,** 301–320. doi:10.1111/j.1467-9868.2005.00503.x (2005).

132. Girosi, F., Jones, M. & Poggio, T. Regularization Theory and Neural Networks Architectures. *Neural Computation* **7,** 219–269. doi:10.1162/neco.1995.7.2.219 (1995).

133. Henderson, H. V. & Searle, S. R. On Deriving the Inverse of a Sum of Matrices. *SIAM Review* **23,** 53–60. doi:10.1137/1023004 (1981).

134. Rasmussen, C. E. & Williams, C. K. I. *Gaussian Processes for Machine Learning* (MIT Press, Cambridge, Mass, 2006).

135. Smola, A. J. & Schölkopf, B. A Tutorial on Support Vector Regression. *Statistics and Computing* **14,** 199–222. doi:10.1023/B:STCO.0000035301.49549.88 (2004).

136. Chang, C.-C. & Lin, C.-J. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology* **2,** 27:1–27:27. doi:10.1145/1961189.1961199 (2011).

137. De Jong, S. & Kiers, H. A. L. Principal Covariates Regression: Part I. Theory. *Chemometrics and Intelligent Laboratory Systems. Proceedings of the 2nd Scandinavian Symposium on Chemometrics* **14,** 155–164. doi:10.1016/0169-7439(92)80100-I (1992).

138. Vervloet, M., Kiers, H. A. L., den Noortgate, W. V. & Ceulemans, E. PCovR: An R Package for Principal Covariates Regression. *Journal of Statistical Software* **65,** 1–14. doi:10.18637/jss.v065.i08 (2015).

139. Fischer, M. J. Regularized Principal Covariates Regression and Its Application to Finding Coupled Patterns in Climate Fields. *Journal of Geophysical Research: Atmospheres* **119,** 1266–1276. doi:10.1002/2013JD020382 (2014).

140. Van Deun, K., Crompvoets, E. A. V. & Ceulemans, E. Obtaining Insights from High-Dimensional Data: Sparse Principal Covariates Regression. *BMC Bioinformatics* **19,** 104. doi:10.1186/s12859-018-2114-5 (2018).

141. Vervloet, M., Van Deun, K., Van den Noortgate, W. & Ceulemans, E. Model Selection in Principal Covariates Regression. *Chemometrics and Intelligent Laboratory Systems* **151,** 26–33. doi:10.1016/j.chemolab.2015.12.004 (2016).

142. Vervloet, M., Van Deun, K., Van den Noortgate, W. & Ceulemans, E. On the Selection of the Weighting Parameter Value in Principal Covariates Regression. *Chemometrics and Intelligent Laboratory Systems* **123,** 36–43. doi:10.1016/j.chemolab.2013.02.005 (2013).

143. Jong, S. D. & Braak, C. J. F. T. Comments on the PLS kernel algorithm. *Journal of Chemometrics* **8,** 169–174. doi:10.1002/cem.1180080208 (1994).

144. Lindgren, F., Geladi, P. & Wold, S. The Kernel Algorithm for PLS. *Journal of Chemometrics* **7,** 45–59. doi:10.1002/cem.1180070104 (1993).

145. Rosipal, R. & Trejo, L. J. Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert Space. *Journal of Machine Learning Research* **2,** 97–123 (2001).

146. Wold, S., Sjöström, M. & Eriksson, L. PLS-Regression: A Basic Tool of Chemometrics. *Chemometrics and Intelligent Laboratory Systems. PLS Methods* **58,** 109–130. doi:10.1016/S0169-7439(01)00155-1 (2001).

147. Lee, M. H. & Liu, Y. Kernel Continuum Regression. *Computational Statistics & Data Analysis* **68,** 190–201. doi:10.1016/j.csda.2013.06.016 (2013).

148. Stone, M. & Brooks, R. J. Continuum Regression: Cross-Validated Sequentially Constructed Prediction Embracing Ordinary Least Squares, Partial Least Squares and Principal Components Regression. *Journal of the Royal Statistical Society. Series B (Methodological)* **52,** 237–269. JSTOR: 2345437 (1990).

149. Hildebrandt, D. & Glasser, D. Predicting Phase and Chemical Equilibrium Using the Convex Hull of the Gibbs Free Energy. *The Chemical Engineering Journal and the Biochemical Engineering Journal* **54,** 187–197. doi:10.1016/0923-0467(94)00202-9 (1994).

150. Anelli, A., Engel, E. A., Pickard, C. J. & Ceriotti, M. Generalized Convex Hull Construction for Materials Discovery. *Physical Review Materials* **2,** 103804. doi:10.1103/PhysRevMaterials.2.103804 (2018).

151. Jolliffe, I. T. A Note on the Use of Principal Components in Regression. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **31,** 300–303. doi:10.2307/2348005. JSTOR: 2348005 (1982).

152. Rosipal, R., Girolami, M. & Trejo, L. J. *Kernel PCA Feature Extraction of Event-Related Potentials for Human Signal Detection Performance* in *Artificial Neural Networks in Medicine and Biology* (eds Malmgren, H., Borga, M. & Niklasson, L.) (Springer, London, 2000), 321–326. doi:10.1007/978-1-4471-0513-8_49.

153. Rosipal, R., Girolami, M., Trejo, L. J. & Cichocki, A. Kernel PCA for Feature Extraction and De-Noising in Nonlinear Regression. *Neural Computing & Applications* **10,** 231–243. doi:10.1007/s521-001-8051-z (2001).

154. Wibowo, A. & Yamamoto, Y. A Note on Kernel Principal Component Regression. *Computational Mathematics and Modeling* **23,** 350–367. doi:10.1007/s10598-012-9143-0 (2012).

155. Hoegaerts, L., Suykens, J. A. K., Vandewalle, J. & De Moor, B. Subset Based Least Squares Subspace Regression in RKHS. *Neurocomputing. New Aspects in Neurocomputing: 11th European Symposium on Artificial Neural Networks* **63,** 293–323. doi:10.1016/j.neucom.2004.04.013 (2005).

156. Jade, A. M. *et al.* Feature Extraction and Denoising Using Kernel PCA. *Chemical Engineering Science* **58,** 4441–4448. doi:10.1016/S0009-2509(03)00340-3 (2003).

157. Späth, H. Algorithm 39 Clusterwise Linear Regression. *Computing* **22,** 367–373. doi:10.1007/BF02265317 (1979).

158. Bair, E., Hastie, T., Paul, D. & Tibshirani, R. Prediction by Supervised Principal Components. *Journal of the American Statistical Association* **101,** 119–137. doi:10.1198/016214505000000628 (2006).

159. Wilderjans, T. F., Vande Gaer, E., Kiers, H. A. L., Van Mechelen, I. & Ceulemans, E. Principal Covariates Clusterwise Regression (PCCR): Accounting for Multicollinearity and Population Heterogeneity in Hierarchically Organized Data. *Psychometrika* **82,** 86–111. doi:10.1007/s11336-016-9522-0 (2017).

160. Helfrecht, B. A., Gasparotto, P., Giberti, F. & Ceriotti, M. Atomic Motif Recognition in (Bio)Polymers: Benchmarks From the Protein Data Bank. *Frontiers in Molecular Biosciences* **6,** 24. doi:10.3389/fmolb.2019.00024 (2019).

161. Desiraju, G. R. & Steiner, T. *The Weak Hydrogen Bond: In Structural Chemistry and Biology* first publ. in paperback. *International Union of Crystallography Monographs on Crystallography* **9** (Oxford University Press, Oxford, 2001).

162. Jeffrey, G. A. & Saenger, W. *Hydrogen Bonding in Biological Structures* (Springer, Berlin, 1991).

163. Arunan, E. *et al.* Defining the Hydrogen Bond: An Account (IUPAC Technical Report). *Pure and Applied Chemistry* **83,** 1619–1636. doi:10.1351/PAC-REP-10-01-01 (2011).

164. McDonald, I. K. & Thornton, J. M. Satisfying Hydrogen Bonding Potential in Proteins. *Journal of Molecular Biology* **238,** 777–793. doi:10.1006/jmbi.1994.1334 (1994).

165. Luzar, A. & Chandler, D. Structure and Hydrogen Bond Dynamics of Water–Dimethyl Sulfoxide Mixtures by Computer Simulations. *The Journal of Chemical Physics* **98,** 8160–8173. doi:10.1063/1.464521 (1993).

166. Luzar, A. & Chandler, D. Effect of Environment on Hydrogen Bond Dynamics in Liquid Water. *Physical Review Letters* **76,** 928–931. doi:10.1103/PhysRevLett.76.928 (1996).

167. Baker, E. N. & Hubbard, R. E. Hydrogen Bonding in Globular Proteins. *Progress in Biophysics and Molecular Biology* **44,** 97–179. doi:10.1016/0079-6107(84)90007-5 (1984).

168. Mezei, M. & Beveridge, D. L. Theoretical Studies of Hydrogen Bonding in Liquid Water and Dilute Aqueous Solutions. *The Journal of Chemical Physics* **74,** 622–632. doi:10.1063/1.440819 (1981).

169. Rahman, A. & Stillinger, F. H. Molecular Dynamics Study of Liquid Water. *The Journal of Chemical Physics* **55,** 3336–3359. doi:10.1063/1.1676585 (1971).

170. Brown, I. D. On the Geometry of O–H···O Hydrogen Bonds. *Acta Crystallographica Section A* **32,** 24–31. doi:10.1107/S0567739476000041 (1976).

171. Frishman, D. & Argos, P. Knowledge-Based Protein Secondary Structure Assignment. *Proteins: Structure, Function, and Bioinformatics* **23,** 566–579. doi:10.1002/prot.340230412 (1995).

172. Ramachandran, G. N., Ramakrishnan, C. & Sasisekharan, V. Stereochemistry of Polypeptide Chain Configurations. *Journal of Molecular Biology* **7,** 95–99. doi:10.1016/S0022-2836(63)80023-6 (1963).

173. Kabsch, W. & Sander, C. Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features. *Biopolymers* **22,** 2577–2637. doi:10.1002/bip.360221211 (1983).

174. Simons, K. T., Kooperberg, C., Huang, E. & Baker, D. Assembly of Protein Tertiary Structures from Fragments with Similar Local Sequences Using Simulated Annealing and Bayesian Scoring functions11Edited by F. E. Cohen. *Journal of Molecular Biology* **268,** 209–225. doi:10.1006/jmbi.1997.0959 (1997).

175. Simons, K. T. *et al.* Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins. *Proteins: Structure, Function, and Bioinformatics* **34,** 82–95. doi:10.1002/(SICI)1097-0134(19990101)34:1<82::AID-PROT7>3.0.CO;2-A (1999).

176. Martin, J. *et al.* Protein Secondary Structure Assignment Revisited: A Detailed Analysis of Different Assignment Methods. *BMC Structural Biology* **5,** 17. doi:10.1186/1472-6807-5-17 (2005).

177. Frishman, D. & Argos, P. Incorporation of Non-Local Interactions in Protein Secondary Structure Prediction from the Amino Acid Sequence. *Protein Engineering, Design and Selection* **9,** 133–142. doi:10.1093/protein/9.2.133 (1996).

178. Andersen, C. A. F., Palmer, A. G., Brunak, S. & Rost, B. Continuum Secondary Structure Captures Protein Flexibility. *Structure* **10,** 175–184. doi:10.1016/S0969-2126(02)00700-1 (2002).

179. Haghighi, H., Higham, J. & Henchman, R. H. Parameter-Free Hydrogen-Bond Definition to Classify Protein Secondary Structure. *The Journal of Physical Chemistry B* **120,** 8566–8570. doi:10.1021/acs.jpcb.6b02571 (2016).

180. Jones, D. T. Protein Secondary Structure Prediction Based on Position-Specific Scoring matrices11Edited by G. Von Heijne. *Journal of Molecular Biology* **292,** 195–202. doi:10.1006/jmbi.1999.3091 (1999).

181. Cuff, J. A. & Barton, G. J. Application of Multiple Sequence Alignment Profiles to Improve Protein Secondary Structure Prediction. *Proteins: Structure, Function, and Bioinformatics* **40,** 502–511. doi:10.1002/1097-0134(20000815)40:3<502::AID-PROT170>3.0.CO;2-Q (2000).

182. Nagy, G. & Oostenbrink, C. Dihedral-Based Segment Identification and Classification of Biopolymers I: Proteins. *Journal of Chemical Information and Modeling* **54,** 266–277. doi:10.1021/ci400541d (2014).

183. Muggleton, S., King, R. D. & Stenberg, M. J. E. Protein Secondary Structure Prediction Using Logic-Based Machine Learning. *Protein Engineering, Design and Selection* **5,** 647–657. doi:10.1093/protein/5.7.647 (1992).

184. Rashid, S., Saraswathi, S., Kloczkowski, A., Sundaram, S. & Kolinski, A. Protein Secondary Structure Prediction Using a Small Training Set (Compact Model) Combined with a Complex-Valued Neural Network Approach. *BMC Bioinformatics* **17,** 362. doi:10.1186/s12859-016-1209-0 (2016).

185. Rost, B. & Sander, C. Improved Prediction of Protein Secondary Structure by Use of Sequence Profiles and Neural Networks. *Proceedings of the National Academy of Sciences* **90,** 7558–7562. doi:10.1073/pnas.90.16.7558. pmid: 8356056 (1993).

186. Rost, B. & Sander, C. Prediction of Protein Secondary Structure at Better than 70% Accuracy. *Journal of Molecular Biology* **232,** 584–599. doi:10.1006/jmbi.1993.1413 (1993).

187. Akkaladevi, S., Katangur, A. K., Belkasim, S. & Pan, Y. *Protein Secondary Structure Prediction Using Neural Network and Simulated Annealing Algorithm* in *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. **2** (2004), 2987–2990. doi:10.1109/IEMBS.2004.1403847.

188. Holley, L. H. & Karplus, M. Protein Secondary Structure Prediction with a Neural Network. *Proceedings of the National Academy of Sciences* **86,** 152–156. doi:10.1073/pnas.86.1.152. pmid: 2911565 (1989).

189. Wood, M. J. & Hirst, J. D. Protein Secondary Structure Prediction with Dihedral Angles. *Proteins: Structure, Function, and Bioinformatics* **59,** 476–481. doi:10.1002/prot.20435 (2005).

190. Zhang, B., Li, J. & Lü, Q. Prediction of 8-State Protein Secondary Structures by a Novel Deep Learning Architecture. *BMC Bioinformatics* **19,** 293. doi:10.1186/s12859-018-2280-5 (2018).

191. Hollingsworth, S. A., Lewis, M. C., Berkholz, D. S., Wong, W.-K. & Karplus, P. A. (φ,ψ)2 Motifs: A Purely Conformation-Based Fine-Grained Enumeration of Protein Parts at the Two-Residue Level. *Journal of Molecular Biology* **416,** 78–93. doi:10.1016/j.jmb.2011.12.022 (2012).

192. Bartók, A. P. & Csányi, G. Gaussian Approximation Potentials: A Brief Tutorial Introduction. *International Journal of Quantum Chemistry* **115,** 1051–1057. doi:10.1002/qua.24927 (2015).

193. Gasparotto, P., Hassanali, A. A. & Ceriotti, M. Probing Defects and Correlations in the Hydrogen-Bond Network of Ab Initio Water. *Journal of Chemical Theory and Computation* **12,** 1953–1964. doi:10.1021/acs.jctc.5b01138 (2016).

194. Watkin, D. Structure Refinement: Some Background Theory and Practical Strategies. *Journal of Applied Crystallography* **41,** 491–522. doi:10.1107/S0021889808007279 (2008).

195. Cooper, R. I., Thompson, A. L. & Watkin, D. J. CRYSTALS Enhancements: Dealing with Hydrogen Atoms in Refinement. *Journal of Applied Crystallography* **43,** 1100–1107. doi:10.1107/S0021889810025598 (2010).

196. Pietropaolo, A., Muccioli, L., Berardi, R. & Zannoni, C. A Chirality Index for Investigating Protein Secondary Structures and Their Time Evolution. *Proteins: Structure, Function, and Bioinformatics* **70,** 667–677. doi:10.1002/prot.21578 (2008).

197. Pietrucci, F. & Laio, A. A Collective Variable for the Efficient Exploration of Protein Beta-Sheet Structures: Application to SH3 and GB1. *Journal of Chemical Theory and Computation* **5,** 2197–2201. doi:10.1021/ct900202f (2009).

198. Kountouris, P. & Hirst, J. D. Prediction of Backbone Dihedral Angles and Protein Secondary Structure Using Support Vector Machines. *BMC Bioinformatics* **10,** 437. doi:10.1186/1471-2105-10-437 (2009).

199. Cock, P. J. A. *et al.* Biopython: Freely Available Python Tools for Computational Molecular Biology and Bioinformatics. *Bioinformatics* **25,** 1422–1423. doi:10.1093/bioinformatics/btp163 (2009).

200. Imbalzano, G. *et al.* Automatic Selection of Atomic Fingerprints and Reference Configurations for Machine-Learning Potentials. *The Journal of Chemical Physics* **148,** 241730. doi:10.1063/1.5024611 (2018).

201. Bernstein, N. *et al. libAtoms/QUIP* https://github.com/libAtoms/QUIP.

202. Pedregosa, F. *et al.* Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research* **12,** 2825–2830 (2011).

203. Knerr, S., Personnaz, L. & Dreyfus, G. *Single-Layer Learning Revisited: A Stepwise Procedure for Building and Training a Neural Network* in *Neurocomputing* (eds Soulié, F. F. & Hérault, J.) (Springer Berlin Heidelberg, 1990), 41–50.

204. Helfrecht, B. A., Semino, R., Pireddu, G., Auerbach, S. M. & Ceriotti, M. A New Kind of Atlas of Zeolite Building Blocks. *The Journal of Chemical Physics* **151,** 154112. doi:10.1063/1.5119751 (2019).

205. D'Alessandro, D. M., Smit, B. & Long, J. R. Carbon Dioxide Capture: Prospects for New Materials. *Angewandte Chemie International Edition* **49,** 6058–6082. doi:10.1002/anie.201000431 (2010).

206. Choi, S., Drese, J. H. & Jones, C. W. Adsorbent Materials for Carbon Dioxide Capture from Large Anthropogenic Point Sources. *ChemSusChem* **2,** 796–854. doi:10.1002/cssc.200900036 (2009).

207. Choi, M. *et al.* Stable Single-Unit-Cell Nanosheets of Zeolite MFI as Active and Long-Lived Catalysts. *Nature* **461,** 246–249. doi:10.1038/nature08288 (2009).

208. Ravi, M., Sushkevich, V. L. & van Bokhoven, J. A. Towards a Better Understanding of Lewis Acidic Aluminium in Zeolites. *Nature Materials* **19,** 1047–1056. doi:10.1038/s41563-020-0751-3 (2020).

209. Baerlocher, C., McCusker, L. B. & Olson, D. H. *Atlas of Zeolite Framework Types* 6th rev. ed (Elsevier, Amsterdam, 2007).

210. Baerlocher, C. & McCusker, L. B. *Database of Zeolites Structures* http://www.iza-structure.org/databases.

211. Li, Y., Yu, J. & Xu, R. Criteria for Zeolite Frameworks Realizable for Target Synthesis. *Angewandte Chemie International Edition* **52,** 1673–1677. doi:10.1002/anie.201206340 (2013).

212. Lin, L.-C. *et al.* In Silico Screening of Carbon-Capture Materials. *Nature Materials* **11,** 633–641. doi:10.1038/nmat3336 (2012).

213. Lupulescu, A. I. & Rimer, J. D. In Situ Imaging of Silicalite-1 Surface Growth Reveals the Mechanism of Crystallization. *Science* **344,** 729–732. doi:10.1126/science.1250984. pmid: 24833388 (2014).

214. Kumar, M., Choudhary, M. K. & Rimer, J. D. Transient Modes of Zeolite Surface Growth from 3D Gel-like Islands to 2D Single Layers. *Nature Communications* **9,** 2129. doi:10.1038/s41467-018-04296-4 (2018).

215. Zhu, X. *et al.* Establishing Hierarchy: The Chain of Events Leading to the Formation of Silicalite-1 Nanosheets. *Chemical Science* **7,** 6506–6513. doi:10.1039/C6SC01295G (2016).

216. Blatov, V. A., Ilyushin, G. D. & Proserpio, D. M. The Zeolite Conundrum: Why Are There so Many Hypothetical Zeolites and so Few Observed? A Possible Answer from the Zeolite-Type Frameworks Perceived As Packings of Tiles. *Chemistry of Materials* **25,** 412–424. doi:10.1021/cm303528u (2013).

217. Sanders, M. J., Leslie, M. & Catlow, C. R. A. Interatomic Potentials for SiO2. *Journal of the Chemical Society, Chemical Communications,* 1271–1273. doi:10.1039/C39840001271 (1984).

218. Schröder, K.-P. *et al.* Bridging Hydrodyl Groups in Zeolitic Catalysts: A Computer Simulation of Their Structure, Vibrational Properties and Acidity in Protonated Faujasites (H-Y Zeolites). *Chemical Physics Letters* **188,** 320–325. doi:10.1016/0009-2614(92)90030-Q (1992).

219. Gale, J. D. & Rohl, A. L. The General Utility Lattice Program (GULP). *Molecular Simulation* **29,** 291–341. doi:10.1080/0892702031000104887 (2003).

220. Smith, J. V. *Structural Classification of Zeolites* in *Mineralogical Society of America Special Paper Number One: International Mineralogical Association Papers and Proceedings of the Third General Meeting* Third General Meeting of the International Mineralogical Association (Mineralogical Society of America, Washington, D. C., 1963), 281–290.

221. Auerbach, S. M., Carrado, K. A. & Dutta, P. K. *Handbook of Zeolite Science and Technology* (M. Dekker, New York, 2003).

222. Le Roux, S. & Jund, P. Ring Statistics Analysis of Topological Networks: New Approach and Application to Amorphous GeS2 and SiO2 Systems. *Computational Materials Science* **49,** 70–83. doi:10.1016/j.commatsci.2010.04.023 (2010).

223. Muraoka, K., Chaikittisilp, W. & Okubo, T. Energy Analysis of Aluminosilicate Zeolites with Comprehensive Ranges of Framework Topologies, Chemical Compositions, and Aluminum Distributions. *Journal of the American Chemical Society* **138,** 6184–6193. doi:10.1021/jacs.6b01341 (2016).

224. Curtis, R. A. & Deem, M. W. A Statistical Mechanics Study of Ring Size, Ring Shape, and the Relation to Pores Found in Zeolites. *The Journal of Physical Chemistry B* **107,** 8612–8620. doi:10.1021/jp027447+ (2003).

225. Zones, S. I. Translating New Materials Discoveries in Zeolite Research to Commercial Manufacture. *Microporous and Mesoporous Materials* **144,** 1–8. doi:10.1016/j.micromeso.2011.03.039 (2011).

226. Stoneham, A. M. & Harding, J. H. Interatomic Potentials in Solid State Chemistry. *Annual Review of Physical Chemistry* **37,** 53–80. doi:10.1146/annurev.pc.37.100186.000413 (1986).

227. Balamane, H., Halicioglu, T. & Tiller, W. A. Comparative Study of Silicon Empirical Interatomic Potentials. *Physical Review B* **46,** 2250–2279. doi:10.1103/PhysRevB.46.2250 (1992).

228. King, S. V. Ring Configurations in a Random Network Model of Vitreous Silica. *Nature* **213,** 1112–1113. doi:10.1038/2131112a0 (1967).

229. Guttman, L. Ring Structure of the Crystalline and Amorphous Forms of Silicon Dioxide. *Journal of Non-Crystalline Solids* **116,** 145–147. doi:10.1016/0022-3093(90)90686-G (1990).

230. Marians, C. S. & Hobbs, L. W. Network Properties of Crystalline Polymorphs of Silica. *Journal of Non-Crystalline Solids* **124,** 242–253. doi:10.1016/0022-3093(90)90269-R (1990).

231. Franzblau, D. S. Computation of Ring Statistics for Network Models of Solids. *Physical Review B* **44,** 4925–4930. doi:10.1103/PhysRevB.44.4925 (1991).

232. *R.I.N.G.S. code* http://rings-code.sourceforge.net/.

233. Grisafi, A. & Ceriotti, M. Incorporating Long-Range Physics in Atomic-Scale Machine Learning. *The Journal of Chemical Physics* **151,** 204105. doi:10.1063/1.5128375 (2019).

234. Grisafi, A., Nigam, J. & Ceriotti, M. Multi-Scale Approach for the Prediction of Atomic Scale Properties. *Chemical Science* **12,** 2078–2090. doi:10.1039/D0SC04934D (2021).

235. Henson, N. J., Cheetham, A. K. & Gale, J. D. Theoretical Calculations on Silica Frameworks and Their Correlation with Experiment. *Chemistry of Materials* **6,** 1647–1650. doi:10.1021/cm00046a015 (1994).

236. Mazur, M. *et al.* Synthesis of 'Unfeasible' Zeolites. *Nature Chemistry* **8,** 58–62. doi:10.1038/nchem.2374 (2016).

237. Fraux, G., Cersonsky, R. K. & Ceriotti, M. Chemiscope: Interactive Structure-Property Explorer for Materials and Molecules. *Journal of Open Source Software* **5,** 2117. doi:10.21105/joss.02117 (2020).

238. Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research* **16,** 321–357. doi:10.1613/jair.953 (2002).

239. Kaufman, S., Rosset, S. & Perlich, C. *Leakage in Data Mining: Formulation, Detection, and Avoidance* in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, New York, NY, USA, 2011), 556–563. doi:10.1145/2020408.2020496.

240. Millman, K. J. & Aivazis, M. Python for Scientists and Engineers. *Computing in Science & Engineering* **13,** 9–12. doi:10.1109/MCSE.2011.36 (2011).

241. Oliphant, T. E. Python for Scientific Computing. *Computing in Science Engineering* **9,** 10–20. doi:10.1109/MCSE.2007.58 (2007).

242. Virtanen, P. *et al.* SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **17,** 261–272. doi:10.1038/s41592-019-0686-2 (2020).

243. Harris, C. R. *et al.* Array Programming with NumPy. *Nature* **585,** 357–362. doi:10.1038/s41586-020-2649-2 (2020).

244. Oliphant, T. E. *Guide to NumPy* 2nd ed. (CreateSpace Independent Publishing Platform, North Charleston, SC, USA, 2015).

245. Van der Walt, S., Colbert, S. C. & Varoquaux, G. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science Engineering* **13,** 22–30. doi:10.1109/MCSE.2011.37 (2011).

246. Hjorth Larsen, A. *et al.* The Atomic Simulation Environment—a Python Library for Working with Atoms. *Journal of Physics: Condensed Matter* **29,** 273002. doi:10.1088/1361-648X/aa680e (2017).

247. Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Computing in Science Engineering* **9,** 90–95. doi:10.1109/MCSE.2007.55 (2007).

248. Wolfram Research, Inc. *Mathematica, Version 11.1* Champaign, IL, 2017.

249. Stukowski, A. Visualization and Analysis of Atomistic Simulation Data with OVITO–the Open Visualization Tool. *Modelling and Simulation in Materials Science and Engineering* **18,** 015012. doi:10.1088/0965-0393/18/1/015012 (2010).

250. Momma, K. & Izumi, F. VESTA 3 for Three-Dimensional Visualization of Crystal, Volumetric and Morphology Data. *Journal of Applied Crystallography* **44,** 1272–1276. doi:10.1107/S0021889811038970 (2011).

251. Humphrey, W., Dalke, A. & Schulten, K. VMD: Visual Molecular Dynamics. *Journal of Molecular Graphics* **14,** 33–38. doi:10.1016/0263-7855(96)00018-5 (1996).

252. Stone, J. *An Efficient Library for Parallel Ray Tracing and Animation* Master's Thesis (Computer Science Department, University of Missouri-Rolla, 1998).

Benjamin A. Helfrecht

Chemin des Alouettes 2A, 1027 Lonay, CH · ben.helfrecht@gmail.com ·
LinkedIn: https://www.linkedin.com/in/benjamin-helfrecht/

Education

École Polytechnique Fédérale de Lausanne (EPFL),
Lausanne, Switzerland — July 2021

Doctor of Sciences in Materials Science and Engineering
Thesis: *Structure–property relationships in complex materials by combining supervised and unsupervised machine learning*

Purdue University, West Lafayette, IN USA — August 2017

Master of Science in Materials Science and Engineering
Thesis: *Evaluation of transition metal dichalcogenide encapsulation to improve copper interconnects: An ab initio study*

Purdue University, West Lafayette, IN USA — May 2016

Bachelor of Science in Materials Science and Engineering
Highest Distinction, Honors College

Core Research Projects

Laboratory of Computational Science and Modeling,
EPFL — August 2017–July 2021

Machine learning material structure and properties
- ➤ Analysis of complex materials using supervised and unsupervised machine learning
- ➤ Development of new techniques and methodologies to extract data-driven insights from materials

Strachan Research Group,
Purdue University — August 2014–August 2017

Properties of copper/transition metal dichalcogenide (TMD) interfaces
- ➤ Density functional theory (DFT) on high-performance computing resources to assess stability, adhesion, and electronic structure of TMDs on Cu(111) surfaces
- ➤ Nudged elastic band (NEB) calculations to assess the efficacy of TMDs as Cu diffusion barriers

Electro-metallization cells for conductive bridging random access memory
- ➤ Description of the energetics of charged Cu defects in amorphous silicon dioxide through DFT to understand the electrochemical reactions in programmable metallization cells

Selected Publications

1. **B. A. Helfrecht**, R. K. Cersonsky, G. Fraux, M. Ceriotti, *Machine Learning: Science and Technology* **1**, 045021 (2020)
2. **B. A. Helfrecht**, R. Semino, G. Pireddu, S. M. Auerbach, M. Ceriotti, *The Journal of Chemical Physics* **151**, 154112 (2019)
3. **B. A. Helfrecht**, P. Gasparotto, F. Giberti, M. Ceriotti, *Frontiers in Molecular Biosciences* **6**, 24 (2019)
4. **B. A. Helfrecht**, D. M. Guzman, N. Onofrio, A. H. Strachan, *Physical Review Materials* **1**, 034001 (2017)
5. T. L. Thornell, **B. A. Helfrecht**, S. A. Mullen, A. Bawiskar, K. A. Erk, *ACS Macro Letters* **3**, 1069–1073 (2014)

For a full list of publications, see https://orcid.org/0000-0002-2260-7183

Presentations

Talks

1. **B. A. Helfrecht***, R. Semino, G. Pireddu, S. M. Auerbach, M. Ceriotti, "Evaluating candidates for new zeolites with machine learning", Annual Meeting of the American Institute of Chemical Engineers (virtual), 2020
2. **B. A. Helfrecht***, R. Cersonsky, G. Fraux, M. Ceriotti, "Leveraging structure and property information for building maps of materials", Annual Meeting of the American Institute of Chemical Engineers (virtual), 2020
3. **B. A. Helfrecht***, R. Semino, G. Pireddu, S. M. Auerbach, M. Ceriotti, "Towards building new zeolites with machine learning", MARVEL Junior Seminar, Lausanne CH, 2020

* Presenting author

Posters

1. **B. A. Helfrecht***, P. Gasparotto, F. Giberti, M. Ceriotti, "Identifying structural patterns with machine learning", CPMD Conference, Lausanne CH, 2019
2. **B. A. Helfrecht**, D. Guzman*, N. Onofrio, A. Strachan, "Ab initio simulations of stability and electronic properties of Cu(111)/transition metal dichalcogenide interfaces", Materials Research Society Spring Meeting and Exhibit, Phoenix, AZ USA, 2017
3. **B. A. Helfrecht***, D. Guzman, N. Onofrio, A. Strachan, "Interactions between copper and transition metal dichalcogenides: A density functional theory study", Materials Research Society Fall Meeting and Exhibit, Boston, MA USA, 2016

*Presenting author

Awards and Honors

John L. Bray Memorial Award (2016)
Charles C. Chappelle Graduate Fellowship (2016)
Barry Goldwater Scholar (2015)
ThinkSwiss Research Scholarship (2015)
Mysore Dayananda Materials Engineering Scholarship (2014, 2015)
Edward J. Sopcak Memorial Scholarship (2013)
Purdue University Trustees' Scholar (2012–2016)
Phi Beta Kappa Honor Society
Tau Beta Pi Engineering Honor Society

Teaching Experience

MSE-421: Statistical Mechanics [English] (2018, 2020)
Statistical mechanics for Master students
CS-119(a): Information, Computation, Communication [French] (2019)
C Programming for Bachelor students
Master project supervision (2017)
Multiple time stepping in molecular dynamics
Master project supervision (2021)
A universal engine for the calculation of structural properties of materials

Technical Skills

Python (advanced);
C, C++ (intermediate); Fortran (basic)
Machine learning, data science, data analysis, data visualization
High performance computing
Density functional theory
Data management

Languages

English (native)
French (intermediate)
Spanish (intermediate)