EPFL

Thèse n° 10 349

Convergence without Convexity:
Sampling, Optimization, and Games

Présentée le 30 septembre 2020

à la Faculté des sciences et techniques de l’ingénieur
Laboratoire de systèmes d’information et d’inférence
Programme doctoral en génie électrique

pour l’obtention du grade de Docteur ès Sciences

par

Ya-Ping HSIEH

Acceptée sur proposition du jury

Prof. A. H. Sayed, président du jury
Prof. V. Cevher, directeur de thèse
Prof. A. Krause, rapporteur
Prof. P. Mertikopoulos, rapporteur
Prof. N. Flammarion, rapporteur

École
polytechnique
fédérale
de Lausanne

2020

If there is a 50-50 chance that something can go wrong,
then 9 times out of 10 it will.
— Paul Harvey

Dedicated to my cat,
who suffered tremendous loneliness due to this thesis.

Acknowledgements

This thesis would not have been possible without the help and support of many people. First and foremost, I would like to thank my advisor, Volkan Cevher, for his constant encouragement and for giving me infinite freedom to explore my research interest. He has also been extremely supportive in my career: he always gives high praises to my work; when our paper is accepted, he advertises for me; when our paper is rejected, he curses for me. He even bought a fantastic coffee machine after he figured out that I exist to turn espressos into theorems (with a high failure rate).

I was honored to have Nicolas Flammarion, Andreas Krause, Panayotis Mertikopoulos, and Ali Sayed as members of my thesis defense committee. I am grateful for their time and valuable suggestions. I am also grateful to Jason Lee, who hosted me at Princeton University for three months, for his sharp insights into interesting problems.

Special thanks to Gosia Baltaian, who always provides the promptest and most effective help: whenever I have need, she comes in like a wind, calls on phone like a truck (in French – that is not easy at all), and sends emails like a phantom. She is really the unsung heroine of the lab.

On the other hand, this thesis would have been much better if some people cease to be amazing and distract my attention.

Panayotis Mertikopoulos, a genuinely fantastic gentleman, has provided critical guidance in my PhD study. The first time we met, he came to my poster session at NIPS, and told me “I think your Theorem XXX is wrong.” That REALLY freaked me out, but it turned out that I was right. Later on, when we started collaborating, he metaphorically taught me that “Your whole perspective is wrong.” This time, I remained convinced to date that he was speaking the ultimate truth.

Ahmet Alacaoglu and Ali Kavis brought me to gym and nurtured my athleticism with their patience. I grew all the way from 58kg to almost 80kg in a year, to the point where my mom was like “why the hell are you photoshopping your body?” when she saw me on camera. I am extremely grateful to their muscles, and I will deadlift 200kg in honor of them, soon.

However, one thing that Ahmet and Ali did not teach me well for gym is how to eat, since we most often just yolo on our food. In this regard, Maria-Luiza Vladarean has brought me the important perspective of food discipline, which later on generalized to other aspects of my life. If you ask Maria what she ate today for lunch, she would reply with “621 kcal of substance, in a balanced mixture of metabolic and anabolic functioning”.

I have had great times with members of the LIONS lab, to whom I would like to express my gratitude: Nadav Hallak for his fun and nice personality, Yurii Malitskyi for his “Oh of course

Riemann hypothesis is trivial." idiosyncrasy, Kamalaruban Parameswaran for being a reliable and kind lab mate, Paul Rolland for his ardent French humor, Fabian Latorre for his carefree lifestyle and amazing coffee beans, Mehmet Fatih Sahin for his patience in working with me, Thomas Sanchez for providing great help with TAing, Thomas Pethick for providing consistent help in experiments, and many other new members of LIONS that keep the youth of our lab.

I am also grateful to a lot of people I met at EPFL: Ilija Bogunovic, a dear friend of mine, warmly invited me to his hometown. In return for his kindness, I puked there in a pub (I'm sorry). Marwa El Halabi, for her mellifluous laughter that you can hear from miles away. Yen-Huan Li, the God of the universe, who taught me a lot before he graduated. Jonathan Scarlett, who serves as my role model in research. Alp Yurtsever, for his warm personality and fabulous cooking skill. Baran Gözcü, who introduced me to the beautiful world of Turkish cuisine.

Finally, I want to express my utmost gratitude towards my family for their unconditional love and support. They voluntarily told me that they can provide full financial support even if my PhD is unpaid. I cannot resist such kind offers, and hence they still didn't realize that EPFL pays nice salary to PhDs. I think that is the true meaning of a family.

Lausanne, 27 Aug 2020 Y.-P. H.

Abstract

Many important problems in contemporary machine learning involve solving highly non-convex problems in sampling, optimization, or games. The absence of convexity poses significant challenges to convergence analysis of most training algorithms, and in some cases (such as min-max games) it is not even known whether common training algorithms converge or not. In this thesis, we aim to partially bridge the gap by

1. Proposing a new sampling framework to transform non-convex problems into convex ones.

2. Characterizing the convergent sets of a wide family of popular algorithms for min-max optimization.

3. Devising provably convergent algorithms for finding mixed Nash Equilibria of infinite-dimensional bi-affine games.

Our theory has several important implications. First, we resolve a decade-old open problem in Bayesian learning via our non-convex sampling framework. Second, our algorithms for bi-affine games apply to the formidably difficult training of generative adversarial networks and robust reinforcement learning, and on both examples we demonstrate promising empirical performance. Finally, our results on min-max optimization lead to a series of negative results for state-of-the-art algorithms, suggesting that one requires fundamentally new tools to advance the theory.

Résumé

De nombreux problèmes importants dans l'apprentissage automatique contemporain nécessitent la résolution de problèmes hautement non-convexes d'échantillonnage, d'optimisation ou de jeux. Malheureusement, l'absence de convexité pose des défis importants à l'analyse de convergence de la plupart des algorithmes de formation, et dans certains cas (comme les jeux min-max), on ne sait même pas si les algorithmes de formation communs convergent ou non. Dans cette thèse, nous visons à combler partiellement ce déficit en

1. Proposant un nouveau cadre d'échantillonnage pour transformer les problèmes non-convexes en problèmes convexes.
2. Caractérisant les ensembles convergents d'une grande famille d'algorithmes populaires pour l'optimisation min-max.
3. Développant des algorithmes à convergence prouvée pour trouver des équilibres de Nash mixtes de jeux bidimensionnels à dimension infinie.

Notre théorie a plusieurs implications importantes : Premièrement, nous résolvons un problème ouvert de dix ans dans l'apprentissage Bayésien via notre cadre d'échantillonnage non-convexe. Deuxièmement, nos algorithmes pour les jeux bi-affines s'appliquent à l'entraînement extrêmement difficile des réseaux antagonistes génératifs et à l'apprentissage par renforcement robuste, et nous démontrons, pour les deux exemples, des performances empiriques prometteuses. Enfin, nos résultats sur l'optimisation min-max conduisent à une série de résultats négatifs pour les algorithmes de pointe, ce qui suggère qu'on a besoin d'outils fondamentalement nouveaux pour faire avancer la théorie.

Contents

Acknowledgements v

Abstract (English/Français/Deutsch) vii

List of figures xiv

List of tables xviii

1 Introduction 3
- 1.1 The three fundamental tasks in machine learning 3
- 1.2 State-of-the-arts: the critical role of *non-convexity* 4
- 1.3 Contributions and organization 5
  - 1.3.1 Review of convex techniques: finite games 5
  - 1.3.2 Exploiting hidden convexity 5
  - 1.3.3 Asymptotic behaviors via dynamical system 6
  - 1.3.4 Convex reformulation of non-convex problems 6

2 Warmup: zero-sum finite games 7
- 2.1 Introduction 7
- 2.2 Preliminaries and notation 8
- 2.3 A family of optimistic mirror descents: classical, robust, and let's be honest . . 9
  - 2.3.1 High-level ideas 10
  - 2.3.2 Optimistic mirror descent 10
  - 2.3.3 Robust optimistic mirror descent 11
  - 2.3.4 Let's be honest: the full framework 12
- 2.4 Experiments 17
  - 2.4.1 Honest setting 18
  - 2.4.2 Adversarial setting 18

3 Mirrored Langevin dynamics 21
- 3.1 Introduction 21
- 3.2 Preliminaries 22
  - 3.2.1 Notation 22
  - 3.2.2 Push-forward and optimal transport 22
- 3.3 Mirrored Langevin dynamics 23

3.3.1 Motivation and algorithm . . . 23
3.3.2 Sampling algorithms on simplex and Latent Dirichlet Allocation (LDA) . 24
3.4 Stochastic mirrored Langevin dynamics . . . 26
3.5 Experiments . . . 28
3.5.1 Synthetic experiment for Dirichlet posterior . . . 28
3.5.2 Latent Dirichlet Allocation with Wikipedia corpus . . . 28

4 Spurious convergence of min-max optimization algorithms 31
4.1 Introduction . . . 31
4.2 Setup and preliminaries . . . 34
4.3 Core algorithmic framework . . . 35
4.3.1 The Robbins–Monro template . . . 35
4.3.2 Specific algorithms . . . 35
4.4 Convergence analysis . . . 37
4.4.1 Continuous vs. discrete time . . . 37
4.4.2 Stochastic approximation . . . 38
4.4.3 Applications and examples . . . 39
4.4.4 Convergence analysis . . . 40
4.5 Spurious attractors: illustrations and examples . . . 41

5 Mixed Nash equilibria of min-max optimization problems 45
5.1 Introduction . . . 45
5.2 Mixed Nash equilibria and infinite-dimensional bi-affine games . . . 46
5.2.1 Preliminary: finite bi-affine games . . . 46
5.2.2 Mixed strategy formulation for min-max games . . . 47
5.3 Infinite-dimensional prox methods . . . 48
5.4 A sampling framework for approximate infinite-dimensional prox methods . . 50
5.4.1 Implementable entropic MD: from probability measure to samples . . . 50
5.4.2 Summarizing samples by averaging: a simple yet effective heuristic . . . 52
5.5 Empirical justification of MixedNE-LD . . . 53
5.5.1 Existing algorithms . . . 53
5.5.2 Degree-2 polynomials: stationary points vs. NE . . . 55
5.5.3 Main result . . . 56
5.5.4 A digression: sampling vs. optimization . . . 57
5.6 Experiment I: generative adversarial networks (GANs) . . . 57
5.6.1 Synthetic data . . . 57
5.6.2 Real data . . . 58
5.7 Experiment II: robust reinforcement learning . . . 59
5.7.1 Off-policy (DDPG) experiments . . . 60
5.7.2 On-Policy (VPG) Experiments . . . 65

6 Conclusion and future work 67
6.1 Summary of the thesis . . . 67
6.2 Future directions . . . 68
6.2.1 Mirrored Langevin dynamics and non-convex sampling . . . 68
6.2.2 Theoretical guarantees of Langevin dynamics for min-max games . . . 68
6.2.3 Limit cycles in min-max games . . . 68

A Appendix for Chapter 2 71
A.1 Equivalence formulations of optimistic mirror descent . . . 71
A.2 Optimistic mirror descent . . . 71
A.3 Robust optimistic mirror descent . . . 75

B Appendix for Chapter 3 77
B.1 Proof of Theorem 3.2 . . . 77
B.1.1 Total variation and relative entropy . . . 77
B.1.2 2-Wasserstein distance . . . 78
B.2 Proof of Lemma 3.1 . . . 80
B.3 Proof of Lemma 3.2 . . . 81
B.4 Proof of Theorem 3.3 . . . 82
B.5 Stochastic gradients for Dirichlet posteriors . . . 82
B.6 More on experiments . . . 83
B.6.1 Synthetic Data . . . 83
B.6.2 Comparison against SGRHMC for Latent Dirichlet Allocation . . . 83

C Appendix for Chapter 4 85
C.1 Asymptotic pseudotrajectories . . . 85
C.1.1 Generalities and preliminaries . . . 85
C.1.2 Proof of Theorem 4.1 . . . 86
C.1.3 Proof of Proposition 4.1 . . . 90
C.2 Convergence analysis . . . 94
C.2.1 Convergence to internally chain-transitives (ICTs) . . . 94
C.2.2 Convergence to attractors . . . 95
C.3 Omitted proofs for Section 4.5 . . . 101
C.3.1 A general criterion for spurious ICT sets in almost bilinear games . . . 101
C.3.2 Proof of spurious ICT sets in Example 4.5.2 . . . 102
C.3.3 Second-order methods in Example 4.5.3 as perturbations . . . 103
C.3.4 Further comparisons . . . 103

D Appendix for Chapter 5 105
D.1 A framework for infinite-dimensional mirror descent . . . 105
D.1.1 A note on the regularity . . . 105
D.1.2 Properties of entropic mirror map . . . 105
D.2 Convergence rates for infinite-dimensional prox methods . . . 112

- D.2.1 Rigorous Statements . . . 112
- D.2.2 Proof of convergence rates for infinite-dimensional mirror descent . . . 113
- D.2.3 Proof of convergence rates for infinite-dimensional Mirror-Prox . . . 115
- D.3 Omitted Pseudocodes in the Main Text . . . 119
- D.4 Algorithms and omitted proofs for Section 5.5 . . . 121
  - D.4.1 Algorithms and hyperparameters . . . 121
  - D.4.2 MixedNE-LD on Example 4.5.2 . . . 122
  - D.4.3 Proof of Theorem 5.3 . . . 122
  - D.4.4 Proof of Theorem 5.4 . . . 124
- D.5 Details and further results for Section 5.6 . . . 124
  - D.5.1 Synthetic Data . . . 126
  - D.5.2 Real Data . . . 130
- D.6 Details and further results for Section 5.7 . . . 133
  - D.6.1 Off-policy (DDPG) experiments: algorithms, hyperparameters, and results 133
  - D.6.2 On-Policy (VPG) experiments: algorithms, and hyperparameters, and results . . . 135
  - D.6.3 Ablation study . . . 135
  - D.6.4 Code . . . 136

Bibliography **171**

Curriculum Vitae **173**

List of Figures

2.1 Honest setting. 18

2.2 Adversarial setting. 19

4.1 Comparison of different RM schemes for bilinear games $F(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}\boldsymbol{y}$, $\boldsymbol{x},\boldsymbol{y} \in \mathbb{R}$. From left to right: (*a*) gradient descent/ascent; (*b*) the mean dynamics (MD); (*c*) extra-gradient. 38

4.2 Spurious limits of min-max optimization algorithms. From left to right: (*a*) (SEG) for (4.8) with $\varepsilon = 0.01$; (*b*) "forsaken solutions" of (SEG); (*c*) "forsaken solutions" of symplectic gradient adjustment (SGA). The red curves present trajectories with different initialization; non-critical ICT sets are depicted in white; the blue curves represent an time-averaged sample orbit. 43

5.1 $F(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}^2\boldsymbol{y}^2 - \boldsymbol{x}\boldsymbol{y}$. The NE is (0,0) with reward value 0. The dashed curve $\boldsymbol{x}\boldsymbol{y} = 0.5$ describe all stationary points that are *not* NE. (a), (b) shows the objective value and the training dynamics when initializing far away from NE. (c), (d) shows the objective value and the training dynamics when $(\boldsymbol{x}_1,\boldsymbol{y}_1)$ is initializing close to NE. 54

5.2 $F(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}y - \boldsymbol{x}^2\boldsymbol{y}^2$. The NE is (0,0) with reward value 0. The dashed curve $\boldsymbol{x}\boldsymbol{y} = 0.5$ are stationary points that are *not* NE. (a), (b) shows the objective value and the training dynamics when initializing far away from NE. (c), (d) shows the objective value and the training dynamics when initializing close to NE. 54

5.3 $F(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}^2\boldsymbol{y}^2$. The NE are represented with the line $\{(\boldsymbol{x},0) \mid \boldsymbol{x}$ arbitrary$\}$ with reward value 0. (a), (b) shows the objective value and the training dynamics when initializing far away from NE. (c), (d) shows the objective value and the training dynamics when initializing close to NE. 54

5.4 Fitting 25 Gaussian mixtures up to 10^5 iterations. Blue dots represent the true distribution and red ones are from the trained generator. 58

5.5 Dataset `LSUN bedroom`, 10^5 iterations. 59

5.6 Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed without adversarial perturbations, on a range of mass values not encountered during training. 61

5.7 Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed without adversarial perturbations, on a range of mass values not encountered during training. 62
5.8 Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed without adversarial perturbations, on a range of friction values not encountered during training. 63
5.9 Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed without adversarial perturbations, on a range of friction values not encountered during training. 64

C.1 ConO with $\lambda = 0.2$ applied to (4.9). 104
C.2 Spurious limits of min-max optimization algorithms from the same initialization. From left to right: (*a*) CGA for (4.9); (*b*) (OG/PEG) for (4.9); (*c*) Algorithms for (4.8).104

D.1 Fitting 8 Gaussian mixtures up to 10^5 iterations. 127
D.2 Fitting the ‘Swiss Roll’ up to 10^5 iterations. 128
D.3 Fitting 25 Gaussian mixtures up to 10^5 iterations. 129
D.4 Learning 25 Gaussian mixtures accross different iterations. 130
D.5 True `MNIST` images and samples generated by different algorithms. 132
D.6 Image generated by RMSProp, Simultaneous and Alternated Extra-Adam, Adam, and MixedNE-LD on the `LSUN bedroom` dataset. 134
D.7 Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of noise probability and mass values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant. 141
D.8 Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of noise probability and mass values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and InvertedPendulum. 142
D.9 Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of noise probability and mass values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant. 143
D.10 Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of noise probability and mass values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and InvertedPendulum. 144

D.11 Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of noise probability and friction values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant. 145

D.12 Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of noise probability and friction values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and Inverted-Pendulum. 146

D.13 Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of noise probability and friction values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant. 147

D.14 Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of noise probability and friction values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and Inverted-Pendulum. 148

D.15 Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of friction and mass values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant. 149

D.16 Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of friction and mass values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and InvertedPendulum. 150

D.17 Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of friction and mass values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant. 151

D.18 Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of friction and mass values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and InvertedPendulum. 152

D.19 Average performance (over 5 seeds) of Algorithm 16, and Algorithm 17 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$ and 0 (training on nominal environment $\rho_0 = 0.2$). The evaluation is performed without adversarial perturbations, on a range of environment parameters not encountered during training. 155

D.20 Ablation study: Average performance (over 5 seeds) of MixedNE-LD (with different β, K_t) under the NR-MDP setting with $\delta = 0.1$ (training on Half-cheetah with relative mass 1). The evaluation is performed on a range of noise probability and mass values not encountered during training. 156
D.21 Ablation study: Average performance (over 5 seeds) of MixedNE-LD (with different β, K_t) under the NR-MDP setting with $\delta = 0.1$ (training on Half-cheetah with relative mass 1). The evaluation is performed without adversarial perturbations, on a range of mass values not encountered during training. 157
D.22 Ablation study: Average performance (over 5 seeds) of MixedNE-LD (with different β, K_t) under the NR-MDP setting with $\delta = 0$ (training on nominal environment $\rho_0 = 0.2$). The evaluation is performed without adversarial perturbations, on a range of environment parameters not encountered during training. 157
D.23 Ablation study: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$ (solid lines) and $\delta = 0$ (dashed lines). The evaluation is performed without adversarial perturbations, on a range of mass values not encountered during training. 158
D.24 Ablation study: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$ (solid lines) and $\delta = 0$ (dashed lines). The evaluation is performed without adversarial perturbations, on a range of friction values not encountered during training. 159
D.25 Ablation study: Average performance (over 5 seeds) of different algorithms under the NR-MDP setting with $\delta = 0.1$ (solid lines) and $\delta = 0$ (dashed lines). The evaluation (after training on the nominal environment $\rho_0 = 0.2$) is performed without adversarial perturbations, on a range of environment parameters not encountered during training. 160
D.26 $\delta = 0.1$ 161
D.27 $\delta = 0$ 161
D.28 HalfCheetah-v2 is trained over 2M steps. Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$ (solid lines) and $\delta = 0$ (dashed lines). The evaluation is performed without adversarial perturbations, on a range of mass values not encountered during training. 161
D.29 $\delta = 0.1$ 162
D.30 $\delta = 0$ 162
D.31 HalfCheetah-v2 is trained over 2M steps. Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$ (solid lines) and $\delta = 0$ (dashed lines). The evaluation is performed without adversarial perturbations, on a range of friction values not encountered during training. 162

List of Tables

2.1 A convergence rate comparison in the context of assumptions. 8

D.1 Hyperparameter setting. “S”, “M”, “L” stands for synthetic data, `MNIST` and `LSUN bedroom`, respectively. MD for `LSUN bedroom` uses a RMSProp preconditioner, so the step-size is the same as one in RMSProp. 125
D.2 Inception Score of generator trained on LSUN dataset. The reported scores are based on the average of 6400 images from each generator. 131
D.3 Common hyperparameters for Algorithm 14 and Algorithm 15, where most of the values are chosen from [DHK+17]. 136
D.4 Exploration-related hyperparameters for Algorithm 14 and Algorithm 15 chosen via grid search (for NR-MDP setting with $\delta = 0.1$). 136
D.5 Exploration-related hyperparameters for Algorithm 14 and Algorithm 15 chosen via grid search (for NR-MDP setting with $\delta = 0$). 137
D.6 Common hyperparameters for Algorithm 16 and Algorithm 17. 137
D.7 Additional hyperparameters for Algorithm 16 and Algorithm 17 chosen via grid search (for NR-MDP setting with $\delta = 0.1$) 137
D.8 Additional hyperparameters for Algorithm 16 and Algorithm 17 chosen via grid search (for NR-MDP setting with $\delta = 0$) 137

Bibliographic Note

This dissertation is based on the following publications:

- Ehsan Asadi Kangarshahi*, Ya-Ping Hsieh*, Mehmet Fatih Sahin, and Volkan Cevher. Let's be honest: An optimal no-regret framework for zero-sum games. *International Conference on Machine Learning*, 2018 [KHSC18].[1]

- Ya-Ping Hsieh, Ali Kavis, Paul Rolland, and Volkan Cevher. Mirrored langevin dynamics. *Advances in Neural Information Processing Systems*, 2018 [HKRC18].

- Ya-Ping Hsieh, Chen Liu, and Volkan Cevher. Finding mixed nash equilibria of generative adversarial networks. *International Conference on Machine Learning*, 2019 [HLC19].

- Parameswaran Kamalaruban, Yu-Ting Huang, Ya-Ping Hsieh, Paul Rolland, Cheng Shi, and Volkan Cevher. Robust reinforcement learning via adversarial training with langevin dynamics. *arXiv preprint*, 2020 [KHH$^+$20].

- Ya-Ping Hsieh, Panayotis Mertikopoulos, and Volkan Cevher. The limits of min-max optimization algorithms: convergence to spurious non-critical sets. *arXiv preprint*, 2020 [HMC20].

Other publications relevant to this dissertation are:

- David Carlson, Edo Collins, Ya-Ping Hsieh, Lawrence Carin, and Volkan Cevher. Preconditioned spectral descent for deep learning. *Advances in Neural Information Processing Systems*, 2015 [CCH$^+$15].

- Yen-Huan Li, Ya-Ping Hsieh, Nissim Zerbib, and Volkan Cevher. A geometric view on constrained m-estimators. *arXiv preprint*, 2015 [LHZC15].

- Alp Yurtsever, Ya-Ping Hsieh, and Volkan Cevher. Scalable convex methods for phase retrieval. *2015IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2015 [YHC15].

---

[1] *Equal contribution

- David Carlson, Ya-Ping Hsieh, Edo Collins, Lawrence Carin, and Volkan Cevher. Stochastic spectral descent for discrete graphical models. *IEEE Journal of Selected Topics in Signal Processing*, 2016 [CHC+16].
- Ashkan Norouzi-Fard, Abbas Bazzi, Ilija Bogunovic, Marwa El Halabi, Ya-Ping Hsieh, and Volkan Cevher. An efficient streaming algorithm for the submodular cover problem. *Advances in Neural Information Processing Systems*, 2016 [NFBB+16].
- Gergely Odor, Yen-Huan Li, Alp Yurtsever, Ya-Ping Hsieh, Quoc Tran-Dinh, Marwa El Halabi, and Volkan Cevher. Frank-wolfe works for non-lipschitz continuous gradient objectives: scalable poisson phase retrieval. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016 [OLY+16].
- Nissim Zerbib, Yen-Huan Li, Ya-Ping Hsieh, and Volkan Cevher. Estimation error of the constrained lasso. *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2016 [ZLHC16].
- Marwa El Halabi, Ya-Ping Hsieh, Bang Vu, Quang Nguyen, and Volkan Cevher. General proximal gradient method: A case for non-euclidean norms. *Technical report*, 2017 [EHHV+17].
- Ya-Ping Hsieh and Volkan Cevher. Dimension-free information concentration via exp-concavity. *Algorithmic Learning Theory*, 2018 [HC18].
- Ya-Ping Hsieh, Yu-Chun Kao, Rabeeh Karimi Mahabadi, Alp Yurtsever, Anastasios Kyrillidis, and Volkan Cevher. A non-euclidean gradient descent framework for non-convex matrix factorization. *IEEE Transactions on Signal Processing*, 2018 [HKM+18].
- Maria-Luiza Vladarean, Ahmet Alacaoglu, Ya-Ping Hsieh, and Volkan Cevher. Conditional gradient methods for stochastically constrained convex minimization. *International Conference on Machine Learning*, 2020 [VAHC20].

1 Introduction

1.1 The three fundamental tasks in machine learning

Modern *machine learning* (ML) can be viewed as a field of *engineering* whose primary goal is to automate decision making based on past observations, i.e., the data. The keyword "engineering" above emphasizes the interdisciplinary nature of ML, which has been strongly influenced by the theory of computer science, statistics, optimization, and control. To illustrate such a perspective, [Jor19] vividly drew an analogy to the development of civil and chemical engineering:

> "Whereas civil engineering and chemical engineering built upon physics and chemistry, this new engineering discipline (ML) will build on ideas that the preceding century gave substance to, such as information, algorithm, data, uncertainty, computing, inference, and optimization."

Despite the vast generality and all-encompassing scope of ML, there are a few fundamental tasks that lie at the heart of almost every single application. First, by the very definition of ML, the ultimate goal is to identify the "best decision" given data. Mathematically, this can be concisely formulated as:

Task 1: Optimization. Given a function $f(\boldsymbol{x})$, find $\boldsymbol{x}^\star := \arg\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$.

Next, due to the ubiquitous noise in data, computing systems, and algorithms, many applications require to incorporate, quantify, and even exploit the *stochasticity* in learning decision rules. In this regard, the most convenient paradigm is given by *sampling*:

Task 2: Sampling. Given a probability distribution $\mathrm{d}\mu(\boldsymbol{x}) \propto e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}$, generate a random variable (called a "sample") $\boldsymbol{X}$ whose distribution is (approximately) $\mathrm{d}\mu$.

Finally, we will focus on an important scenario in modern ML where each learner no longer makes decision in isolation. Instead, multiple agents are simultaneously involved in a single task, and the "optimal decision" of one agent will invariably depend on all others'. As a simple example, suppose, on a beautiful Sunday, we ask Google map to take us from Geneva to Paris by car. In terms of avoiding traffic, it will be highly suboptimal for Google to recommend everyone the same road, even though it might be the shortest path connecting the two cities. The very same principle applies to most of the recommendation systems (restaurants, movies, etc.), which are foreseen to be a major application of ML in the coming decade.

Mathematically, multi-agent learning is best described as a generic *game* in the sense of [vN28, Nas50]. In this thesis, however, we will particularize to the special case of *min-max games*, which is synonymous to *two-player zero-sum* games. Our purpose is three-fold:

1. Two-player games already capture many of the fundamental challenges of multi-player games.
2. The theoretical understanding of games are limited even for two-player games.
3. Min-max optimization naturally arises also in a number of other important applications, such as *generative adversarial network* (GAN) [GPAM+14], robust reinforcement learning [PDSG17], and adversarial training [MMS+18].

Formally, our final fundamental task is:

Task 3: Min-max games. Let $\boldsymbol{x}, \boldsymbol{y}$ be the decision variables for two players and $F(\boldsymbol{x}, \boldsymbol{y})$ be the loss (resp. reward) function for the $\boldsymbol{x}$ (resp. $\boldsymbol{y}$) player. Find a *saddle-point* (also known as an *equilibrium* [vN28, Nas50]) of the min-max objective:

$$\min_{\boldsymbol{x} \in \mathcal{X}} \max_{\boldsymbol{y} \in \mathcal{Y}} F(\boldsymbol{x}, \boldsymbol{y}).$$

1.2 State-of-the-arts: the critical role of *non-convexity*

All the tasks mentioned above are classical and have been studied for at least a century. However, what sets modern ML apart from these classical inspection is the assumptions on the objectives (f, V, and F). Specifically, the distinguishing feature is that we no longer assume *convexity*.

Take **Task 1** as an illustration. Until a decade ago, the most intensely studied subjects of optimization (at least in ML) were linear and convex programming. Then came 2012, the year which witnessed the advent of AlexNet [KSH12] and its astounding accuracy in image classification, marking the dawn of the “deep learning revolution”. Shortly after, it was recognized that the success of deep learning is in large part due to the ability of empirically solving *non-convex* optimization problems. Nowadays, non-convex optimization has become one of the most active and impactful areas in ML, and many conditions are proposed so that non-convex minimization *provably* converges.

Similar to optimization, the scopes of sampling and min-max games haven been strongly influenced by deep learning, and hence give rise to inherently non-convex tasks. In addition, there exist several classical problems in sampling that are also non-convex (meaning V in **Task 2** is non-convex) and have remained open for decades. Collectively, **Tasks 1-3** with non-convex f, V and non-convex/non-concave F present the most pivotal and pressing challenges in contemporary ML, both in theory and practice.

1.3 Contributions and organization

This thesis is devoted to solving certain important non-convex problems in modern ML. We stress that we put equal emphasis in theory and practice, and therefore by "solving" we mean proposing a provably convergent solution that either works well empirically, or provides immediate practical implications.

We now summarize the contributions as follows.

1.3.1 Review of convex techniques: finite games

Before diving into non-convex problems, we will first revisit *finite* games in Chapter 2. This is because finite games are themselves *sampling* problems from finite distributions, which can be solved via online (convex) optimization techniques. As such, they lie at the intersection of **Tasks 1-3** and serve as a great illustration of the chapters to come.

In Chapter 2, we will resolve an open problem of simultaneously achieving the optimal honest and adversarial regret via modifying a key algorithmic idea, the *mirror descent* (MD). We will also take the opportunity to set up the notations.

Next, we turn to non-convex problems. In full generality, it is well-known that **Tasks 1-3** are NP-Hard. Fortunately, many important applications in ML do admit rich structures that allow for elegant solutions. In this thesis, we will demonstrate three approaches to solving non-convex problems:

1.3.2 Exploiting hidden convexity

While a problem, given in its most natural formulation, might seem non-convex, it is possible that an alternative viewpoint is able to reveal the *hidden convexity*.

In Chapter 3, we show that this is actually the case for a decades-old open problem in Bayesian learning, the *Latent Dirichlet Allocation* (LDA) [BNJ03]. LDA is of paramount importance in Bayesian learning, and in particular presents a powerful paradigm for text modeling. Unfortunately, LDA corresponds to **Task 2** with a non-convex V. In spite of tremendous efforts, LDA is still devoid of an efficiently sampler which possesses convergence guarantees.

Our work [HKRC18] resolves this open problem by showing that LDA is in fact a *convex* sampling task in disguise, whose convexity is unveiled only when we look at the corresponding *dual* distribution, not the original one. As a result, we prove that LDA is essentially as easy as convex sampling problems, for which a plethora of algorithms with strong convergence guarantees exist.

1.3.3 Asymptotic behaviors via dynamical system

Our second approach is to reduce the long-term behaviors of practical, *discrete* algorithms to their *continuous-time* counterparts, which greatly simplifies the analysis.

The most general framework to enact such an approach is via theory of *dynamical systems* [Ben99]. In Chapter 4, we show that, under mild assumptions on the objective F and step-size policies, most algorithms proposed for solving non-convex/non-concave **Task 3** exhibit exactly the same asymptotic behavior. Furthermore, we rigorously prove that such a behavior is in fact undesirable, in the sense that these state-of-the-art algorithms collectively *fail* on some simple, polynomial objectives.

1.3.4 Convex reformulation of non-convex problems

Our final approach for solving non-convex problems is to propose a convex alternative which is equally meaningful as the original one.

The best instantiation of this is approach is von Neumann's formulation of a two-player, finite game [vNM44]: whereas pure strategy equilibrium might seem the most intuitive concept from the first glance, it is well-known to be ill-defined and hence lead to imprecise mathematical questions. Instead, one should consider the *mixed* equilibrium, which is always well-defined by von Neumann's min-max principle.

In Chapter 5, we show that this very same strategy can be carried out to solve challenging min-max optimization problems that are non-convex/non-concave. Specifically, given any non-convex/non-concave min-max problem, we show how to extend von Neumann's min-max principle, and then solve the mixed equilibrium formulation using *infinite-dimensional* convex optimization techniques.

2 Warmup: zero-sum finite games

This chapter resolves an open problem in finite games where we aim to design an algorithm that simultaneously achieves the optimal honest regret, adversarial regret, and convergence to the game value.

Although the problem is convex and online, our core techniques for the solution, in particular the algorithmic idea of *mirror descent* (MD), will continue to serve as the backbone of the chapters to come.

2.1 Introduction

An (offline) two-player zero-sum game with payoff matrix A refers to the solving the min-max problem:

$$V := \min_{\boldsymbol{y}\in\Delta_n} \max_{\boldsymbol{x}\in\Delta_m} \langle \boldsymbol{x}, A\boldsymbol{y} \rangle \tag{2.1}$$

where Δ_d is a d-dimensional simplex, representing the mixed strategies over d actions. The quantity V in (2.1) is called the *value* of the game. Any pair $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ attaining the game value is called an equilibrium strategy.

In this chapter, we are interested in the decentralized setting (aka., the "strongly uncoupled" setting), where the payoff matrix and the number of opponent's strategies are unknown to both players, and their goal is to learn a pair of equilibrium strategy through repeated game plays. Moreover, each player aims to suffer a low individual regret, even in the presence of an adversary or a corrupted channel that distorts the feedback. Such a setting is of great interest in optimization and behavioral economics [Mye99], especially under communication constraints.

Specifically, at each round t, the players take actions $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, and then receive the loss vectors $-A\boldsymbol{y}_t$ (for $\boldsymbol{x}$-player) and $A^\top\boldsymbol{x}_t$ (for $\boldsymbol{y}$-player). In the honest setting, we assume that the two players take actions according to a prescribed algorithm, and we say the setting is

	Honest R_T	**Adversarial** R_T	**Game Value**	**Oracle**	**Algorithm**
[DDK11]	$O(\log T)$	$O(\sqrt{T})$	$O(T^{-1}\log^{\frac{3}{2}} T)$	$\lvert A\rvert_{\max}$	Complicated
[RS13b]	?	$O(\sqrt{T}\log T)$	$O\big(T^{-1}\log T\big)$	$T, \lvert A\rvert_{\max}$	Simple
This chapter	$O(\log T)$	$O(\sqrt{T})$	$O\big(T^{-1}\big)$	$\lvert A\rvert_{\max}$	Simple

Table 2.1: A convergence rate comparison in the context of assumptions.

adversarial if only one player (the $\boldsymbol{x}$-player in this chapter) adheres to the prescribed algorithm and the other player arbitrary.

As in previous work, we assume that an upper bound $|A|_{\max}$ on the maximum absolute entry of A is available to both players. The goal is to achieve

$$\begin{aligned} \big|V - \langle \boldsymbol{x}_T, A\boldsymbol{y}_T\rangle\big| &\le r_1(T), \\ R_T := \max_{\boldsymbol{x}\in\Delta_m} \sum_{t=1}^{T} \langle \boldsymbol{x}_t - \boldsymbol{x}, -A\boldsymbol{y}_t\rangle &\le r_2(T) \end{aligned}$$

for fast-decaying r_1 and sublinear r_2 in T. The first requirement is to approximate the game value in (2.1), and the second one asks to minimize the regret R_T.

An open problem proposed by [DDK11] posits the existence of a simple algorithm that converges at optimal rates for both regret and the value of the game in an uncoupled manner, both against honest (i.e., cooperative) and dishonest (i.e., arbitrarily adversarial) behavior. The chapter precisely resolves this challenge:

Theorem 2.1 (Main result of Chapter 2, informal)**.** *For* (2.1)*, there is a simple decentralized algorithm with non-adaptive step-size such that*

$$r_1(T) = O\left(\frac{1}{T}\right), \qquad r_2(T) = O\big(\log T\big),$$

if the opponent is honest (i.e., playing collaboratively to solve the game). Moreover, against any adversary, we have

$$r_2(T) = O\left(\sqrt{T}\right).$$

Except for the $O\big(\log T\big)$ honest regret, these rates are known to be optimal [CBL06, DDK15]. We are also the first to remove $\log T$ factors in convergence to the value of the game, an open question posed by the very first work in learning decentralized games [DDK11]. An comparison of our rates against the prior arts is summarized in Table 2.1.

2.2 Preliminaries and notation

Let h be a function over the convex domain $\mathcal{D}$ that is 1-strongly convex with respect to the norm $\|\cdot\|$, and let $D(\cdot,\cdot)$ be the Bregman divergence associated with h. We will make heavy use of the *three-point identity* for Bregman divergence in the sequel:

$$D(\boldsymbol{x},\boldsymbol{y})+D(\boldsymbol{y},\boldsymbol{z})=D(\boldsymbol{x},\boldsymbol{z})+\left\langle \boldsymbol{x}-\boldsymbol{y},\nabla h(\boldsymbol{z})-\nabla h(\boldsymbol{y})\right\rangle.$$

We use the notation $\boldsymbol{z}=\mathrm{MD}_\eta(\boldsymbol{x},\boldsymbol{g})$ to denote:

$$\boldsymbol{z}=\nabla h^\star\left(\nabla h(\boldsymbol{x})-\eta\boldsymbol{g}\right)$$

where $h^\star$ is the Fenchel dual of h.

We define

$$D^2:=\max\left\{\sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{D}}\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{x}'\|^2,\sup_{\boldsymbol{x}\in\mathcal{D}}D(\boldsymbol{x},\boldsymbol{x}_c)\right\}$$

where $\boldsymbol{x}_c:=\arg\min_{\boldsymbol{x}\in\mathcal{D}}h(\boldsymbol{x})$ is the prox center. Hence D controls both the diameter (in $\|\cdot\|$) and the Bregman divergence to the prox center.

We frequently use the fact that

$$\left\langle \boldsymbol{x},A\boldsymbol{y}\right\rangle\le|A|_{\max}\quad\forall\boldsymbol{x}\in\Delta_m,\ \boldsymbol{y}\in\Delta_n$$

where $|A|_{\max}$ is the maximum entry of A in absolute value, and $\Delta_m:=\{\boldsymbol{x}\in\mathbb{R}^m\mid\sum_{i=1}^m x_i=1,x_i\ge 0\}$ is the standard simplex. On a simplex, we will only consider the entropic mirror map:

$$h(\boldsymbol{x})=\sum_{i=1}^k x_i\log x_i,\quad k=m\text{ or }n \tag{2.2}$$

which is well-known to be 1-strongly convex in $\|\cdot\|_1$.

We use $\frac{1}{m}1_m$ to denote the uniform distribution on Δ_m.

2.3 A family of optimistic mirror descents: classical, robust, and let's be honest

We first illustrate the high-level ideas to prove Theorem 2.1 in Section 2.3.1. A novel analysis for OMD in the honest setting is given in Section 2.3.2, and we propose a new algorithm for the adversarial setting in Section 2.3.3. Finally, the full algorithm is presented in Section 2.3.4, along with the rigorous version of the main result (cf. Theorem 2.4).

Algorithm 1: Optimistic Mirror Descent: $\boldsymbol{x}$-Player

```
Set $\eta = \frac{1}{2|A|_{\max}}$
Play $\boldsymbol{z}_1 = \boldsymbol{z}_2 = \boldsymbol{z}_3 = \frac{1}{m}1_m$
For $t \geq 3$:
    Compute $\boldsymbol{x}_{t+1} = \mathrm{MD}_\eta(\boldsymbol{x}_t, -2(t-2)A\boldsymbol{w}_t + 3(t-3)A\boldsymbol{w}_{t-1} - (t-4)A\boldsymbol{w}_{t-2})$
    Play $\boldsymbol{z}_{t+1} = \frac{1}{t-1}\sum_{i=3}^{t+1}\boldsymbol{x}_i$
    Observe $-A\boldsymbol{w}_{t+1}$
```

Algorithm 2: Optimistic Mirror Descent: $\boldsymbol{y}$-Player

```
Set $\eta = \frac{1}{2|A|_{\max}}$
Play $\boldsymbol{w}_1 = \boldsymbol{w}_2 = \boldsymbol{w}_3 = \frac{1}{n}1_n$
For $t \geq 3$:
    Compute $\boldsymbol{y}_{t+1} = \mathrm{MD}_\eta(\boldsymbol{y}_t, 2(t-2)A^\top\boldsymbol{z}_t - 3(t-3)A^\top\boldsymbol{z}_{t-1} + (t-4)A^\top\boldsymbol{z}_{t-2})$
    Play $\boldsymbol{w}_{t+1} = \frac{1}{t-1}\sum_{i=3}^{t+1}\boldsymbol{y}_i$
    Observe $A^\top\boldsymbol{z}_{t+1}$
```

2.3.1 High-level ideas

Our algorithms are inspired by the iterates of the form:

$$\begin{cases} \boldsymbol{x}_{t+1} = \mathrm{MD}_\eta(\boldsymbol{x}_t, -2A\boldsymbol{y}_t + A\boldsymbol{y}_{t-1}) \\ \boldsymbol{y}_{t+1} = \mathrm{MD}_\eta(\boldsymbol{y}_t, 2A^\top\boldsymbol{x}_t - A^\top\boldsymbol{x}_{t-1}) \end{cases}, \tag{2.3}$$

which are equivalent to the OMD in [RS13b] (see Appendix A.1). It is known that directly applying (2.3) to (2.1) yields $O\left(\frac{1}{T}\right)$ convergence in the game value, however without any guarantee on the regret.

To make OMD optimal for zero-sum games, we improve (2.3) on two fronts. First, in the honest setting, we make the following simple observation: although the iterates $\boldsymbol{x}_t$ are not guaranteed to possess sublinear regret, the averaged iterates $\frac{1}{t}\sum_{i=1}^{t}\boldsymbol{x}_i$ do enjoy logarithmic regret, and hence, it suffices to play the averaged iterates in the honest setting.

Second, in order to make OMD robust against any adversary, we utilize the “mixing steps” of [RS13b] with an important improvement: our step-sizes do not depend on the time horizon. This new feature is crucial in removing $\log T$ factors in both the convergence to game value and adversarial regret. In fact, our analysis is arguably simpler than [RS13b].

2.3.2 Optimistic mirror descent

As alluded to in Section 2.3.1, we will play OMD with the averaged iterates. The algorithms are given explicitly in Algorithms 1–2.

Remark 2.3.1. Note that there is no need to play $\frac{1}{m}1_m$ and $\frac{1}{n}1_n$ three times in Algorithms 1–2. The players could just play once $\left(\frac{1}{m}1_m\right)^\top A\left(\frac{1}{n}1_n\right)$ and would have enough information to run OMD from $\boldsymbol{x}_4$ and $\boldsymbol{y}_4$. Our choices are motivated by the resulting ease of the notation.

We analyze our version of OMD below. The crux of our analysis is to first look at the regrets of auxiliary sequences $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, and we show that the *sum* of the auxiliary regrets, not any individual of them, controls both the convergence to the value of the game and the honest regret for the averaged sequences $\boldsymbol{z}_t$ and $\boldsymbol{w}_t$.

Theorem 2.2. *Suppose two players of a zero-sum game have played T rounds according to the OMD algorithm with $\eta = \frac{1}{2|A|_{\max}}$. Then*

1. *The $\boldsymbol{x}$-player suffers an $O\big(\log T\big)$ regret:*

$$\max_{\boldsymbol{z}\in\Delta_m}\sum_{t=3}^{T}\langle \boldsymbol{z}_t-\boldsymbol{z},-A\boldsymbol{w}_t\rangle \le \log 2(T-2)|A|_{\max}\times\Big(20+\log m+\log n\Big) \tag{2.4}$$
$$= O\big(\log T\big)$$

   and similarly for the $\boldsymbol{y}$-player.

2. *The strategies $(\boldsymbol{z}_T,\boldsymbol{w}_T)$ constitutes an $O\big(\frac{1}{T}\big)$-approximate equilibrium to the value of the game:*

$$|V-\langle \boldsymbol{z}_T,A\boldsymbol{w}_T\rangle| \le \frac{\Big(20+\log m+\log n\Big)|A|_{\max}}{T-2} \tag{2.5}$$
$$= O\left(\frac{1}{T}\right).$$

Proof. See Appendix A.2. ■

2.3.3 Robust optimistic mirror descent

In this section, we introduce the *robust optimistic mirror descent* (ROMD), which is a novel algorithm even for online convex optimization.

Let h be 1-strongly convex with respect to $\|\cdot\|$, and suppose we are minimizing the regret against an arbitrary sequence of convex functions $f_1, f_2, \ldots$ in a constraint set $\mathcal{D}$. Assume that each function is G-Lipschitz in $\|\cdot\|$. Assume also that no Bregman projection is needed (i.e., $\mathrm{MD}_\eta(\boldsymbol{x},\boldsymbol{g})\in\mathcal{D}$ for any $\boldsymbol{x}$ and $\boldsymbol{g}$); this is, for instance, the case for the entropic mirror map.

We state ROMD in the general form in Algorithm 3.

Theorem 2.3 ($O(\sqrt{T})$-Adversarial Regret)**.** *Suppose that $\|\nabla f_t\|_* \le G$ for all t. Then playing T rounds of Algorithm 3 with $\eta_t = \frac{1}{G\sqrt{t}}$ against an arbitrary sequence of convex functions has the following guarantee on the regret:*

$$\max_{\boldsymbol{x}\in\Delta_m}\sum_{t=1}^{T}\big\langle \boldsymbol{x}_t-\boldsymbol{x},\nabla f_t(\boldsymbol{x}_t)\big\rangle \le G\sqrt{T}\big(18+2D^2\big)+GD\Big(3\sqrt{2}+4D\Big)$$
$$= O\Big(\sqrt{T}\Big).$$

Algorithm 3: Robust Optimistic Mirror Descent

```
Initialize $\boldsymbol{x}_1 = \boldsymbol{x}_c$, $\nabla f_0 = 0$, $\eta_t = \frac{1}{G\sqrt{t}}$
for $t = 1, 2, \ldots,$ do
    $\tilde{\boldsymbol{x}}_t = (\frac{t-1}{t})\boldsymbol{x}_t + \frac{1}{t}\boldsymbol{x}_c$
    Set $\tilde{\nabla}_t = 2\nabla f_t(\boldsymbol{x}_t) - \nabla f_{t-1}(\boldsymbol{x}_{t-1})$,
play $\boldsymbol{x}_{t+1} = \mathrm{MD}_{\eta_t}(\tilde{\boldsymbol{x}}_t, \tilde{\nabla}_t)$
    Observe $f_{t+1}$
end for
```

Proof. See Appendix A.3. ■

When specialized to zero-sum games, it suffices to take $\boldsymbol{x}_c = \frac{1}{m}1_m$, $G = |A|_{\max}$, $D = \log m$, and h being the entropic mirror map.

2.3.4 Let's be honest: the full framework

We now present our approach for solving (2.1).

To ease the notation, define

$$\boldsymbol{z}_t^* := \arg\min_{\boldsymbol{x}\in\Delta_m} \langle \boldsymbol{x}, -A\boldsymbol{w}_t \rangle$$

and

$$\boldsymbol{w}_t^* = \arg\min_{\boldsymbol{y}\in\Delta_n} \left\langle \boldsymbol{z}_t, A\boldsymbol{y} \right\rangle.$$

Let constants C_1, C_2, and C_3 be such that (see Theorems 2.2–2.3 and (A.14))

$$\left\langle \boldsymbol{z}_t - \boldsymbol{z}_t^*, -A\boldsymbol{w}_t \right\rangle \le \frac{C_1}{t}, \quad \boldsymbol{z}_t, \boldsymbol{w}_t \text{ from OMD,} \tag{2.6}$$

$$\left\langle \boldsymbol{w}_t - \boldsymbol{w}_t^*, A^\top \boldsymbol{z}_t \right\rangle \le \frac{C_1}{t}, \quad \boldsymbol{z}_t, \boldsymbol{w}_t \text{ from OMD,} \tag{2.7}$$

$$\sum_{t=1}^{T} \left\langle \boldsymbol{z}_t - \boldsymbol{z}^*, -A\boldsymbol{y}_t \right\rangle \le C_2\sqrt{T}, \quad \boldsymbol{z}_t \text{ from ROMD and } \boldsymbol{y}_t \text{ arbitrary,} \tag{2.8}$$

$$|V - \boldsymbol{z}_T A\boldsymbol{w}_T| \le \frac{C_3}{T}, \quad \boldsymbol{z}_T, \boldsymbol{w}_T \text{ from OMD.} \tag{2.9}$$

From a high-level, our approach exploits the following simple observation: suppose that we know C_1 above. If the instantaneous regret bound (2.6) and (2.7) hold true for all t, then we would trivially have the desired convergence.

In contrast, if at any round the bound (2.6) is violated for the $\boldsymbol{x}$-player, then it must be due to an adversarial play, and we can simply switch to ROMD to get $O(\sqrt{T})$ regret. However, since C_1 (*cf.*, (A.14)) involves n, the number of opponent's strategies, the $\boldsymbol{x}$-player cannot compute

Algorithm 4: Let's Be Honest Optimistic Mirror Descent: $\boldsymbol{x}$-Player

```
1:  Initialize $b = 1, t = 1, \boldsymbol{w}_0 = \frac{1}{n}1_n$ and $\boldsymbol{z}_0 = \frac{1}{m}1_m$
2:  Play $t$-th round of OMD-$\boldsymbol{x}$, observe $-A\mathbf{p}_t$
3:  if $G_t^{\boldsymbol{w}} := \langle \boldsymbol{w}_{t-1}, A^\top \boldsymbol{z}_{t-1} \rangle - \langle \mathbf{p}_t, A^\top \boldsymbol{z}_{t-1} \rangle > \frac{b}{t-1}$ then
4:      Play $b^4 - 1$ rounds of ROMD
5:      $t \leftarrow t+1$
6:      $b \leftarrow 2b$
7:      Go to line 2.
8:  end if
9:  $-A\boldsymbol{w}_t \leftarrow -A\mathbf{p}_t$
10: if $G_t^{\boldsymbol{z}} := \langle \boldsymbol{z}_t, -A\boldsymbol{w}_t \rangle - \langle \boldsymbol{z}_t^*, -A\boldsymbol{w}_t \rangle > \frac{b}{t}$ then
11:     Play $\check{\boldsymbol{x}}_{t+1} := \boldsymbol{z}_t^*$
12:     Play $b^4 - 1$ rounds of ROMD
13:     $t \leftarrow t+2$
14:     $b \leftarrow 2b$
15:     Go to line 2.
16: end if
17: $t \leftarrow t+1$
18: Go to line 2.
```

it exactly. The situation is similar for the $\boldsymbol{y}$-player. We hence need to come up with a way to estimate C_1 for both players.

It is important to note that one can not naïvely estimate C_1 by binary search separately on both players. The reason, and the major difficultly to the above approach, is as follows: since in general $\langle \boldsymbol{z}_t - \boldsymbol{z}_t^*, -A\boldsymbol{w}_t \rangle \neq \langle \boldsymbol{w}_t - \boldsymbol{w}_t^*, A^\top \boldsymbol{z}_t \rangle$, it could be the case that, at the same round, the $\boldsymbol{x}$-player detects a bad instantaneous regret and switch to ROMD, while the $\boldsymbol{y}$-player remains in OMD, even though two players are both honest. However, our entire analysis of OMD would breakdown if the OMD is not played cohesively.

Furthermore, recall that we also want robustness against any adversary. Therefore, a bad instantaneous regret indicates the possibility of receiving an adversarial play, and we need to switch to ROMD whenever this occurs.

To resolve such issues, we devise a simple *signaling* scheme ($\check{\boldsymbol{x}}_t$ and $\check{\boldsymbol{y}}_t$ in Algorithms 4–5), which synchronizes both players' C_1 estimate and also the OMD plays while guaranteeing robustness.

In words, our signaling scheme is a "Let's be honest" message to the opponent: "I am having a bad instantaneous regret. Please update your C_1 with me, and please pretend that I am adversarial for a small number of rounds, so that we can play honest OMD cohesively." It turns out that doing these extra signaling rounds do not hurt the convergence rates in OMD and ROMD at all.

Our full algorithm, termed *Let's Be Honest* (LbH) *Optimistic Mirror Descent*, is presented in

Algorithm 5: Let's Be Honest Optimistic Mirror Descent: $\boldsymbol{y}$-Player

```
1: Initialize $b = 1, t = 1, \boldsymbol{w}_0 = \frac{1}{n}1_n$ and $\boldsymbol{z}_0 = \frac{1}{m}1_m$
2: Play $t$-th round of OMD-$\boldsymbol{y}$, observe $A^\top \mathbf{o}_t$
3: if $G_t^{\boldsymbol{z}} := \langle \boldsymbol{z}_{t-1}, -A\boldsymbol{w}_{t-1}\rangle - \langle \mathbf{o}_t, -A\boldsymbol{w}_{t-1}\rangle > \frac{b}{t-1}$ then
4:         Play $b^4 - 1$ rounds of ROMD
5:         $t \leftarrow t+1$
6:         $b \leftarrow 2b$
7:         Go to line 2.
8: end if
9: $A\boldsymbol{z}_t \leftarrow A^\top \mathbf{o}_t$
10: if $G_t^{\boldsymbol{w}} := \langle \boldsymbol{w}_t, A^\top \boldsymbol{z}_t\rangle - \langle \boldsymbol{w}_t^*, A^\top \boldsymbol{z}_t\rangle > \frac{b}{t}$ then
11:         Play $\check{\boldsymbol{y}}_{t+1} := \boldsymbol{w}_t^*$
12:         Play $b^4 - 1$ rounds of ROMD
13:         $t \leftarrow t+2$
14:         $b \leftarrow 2b$
15:         Go to line 2.
16: end if
17: $t \leftarrow t+1$
18: Go to line 2.
```

Algorithms 4–5.

Remark 2.3.2. In Algorithms 4–5, the role of b is to estimate the constant C_1 in (2.6). Since our analysis requires b to be the same for both players throughout the algorithm run, a simple way is to assume that, say, $m = n = 5$, compute the corresponding $\tilde{C}_1$, and set the initial $b \leftarrow \tilde{C}_1$. Doing so indeed improves upon constants in our convergence; we chose $b = 1$ only for simplicity.

Remark 2.3.3. There are some degree of freedom in Algorithms 4–5. For instance, instead of doubling b in Line 16, one can do $b \leftarrow (1+\epsilon)b$ for some $\epsilon > 0$. In Line 5, one can also play $b^2 - 1$ rounds, rather than $b^4 - 1$. As will become apparent in Theorem 2.4, these variants only effect the constants but not the convergence rates. However, they do have impact on empirical performance; cf. Section 2.4.

The following key lemma ensures the two players to enter the ROMD plays coherently.

Lemma 2.1. *If the $\boldsymbol{y}$-player enters Line 12 of Algorithm 5 at the t-th round, then the $\boldsymbol{x}$-player enters Line 4 of Algorithm 4 at the $(t+1)$-th round. Conversely, if, at the t-th round, the $\boldsymbol{y}$-player does not enter Line 12 of Algorithm 5, then the $\boldsymbol{x}$-player does not enter Line 4 of Algorithm 4 at the $(t+1)$-th round.*

Exactly the same statements hold when the $\boldsymbol{x}$- and $\boldsymbol{y}$-player are reversed above.

Proof. If the $\boldsymbol{y}$-player enters Line 12 of Algorithm 5 at the t-th round, then $\check{\boldsymbol{y}}_{t+1}$ is signalled at the $(t+1)$-th round, and it must be the case that $\langle \boldsymbol{w}_t - \boldsymbol{w}_t^*, A^\top \boldsymbol{z}_t\rangle > \frac{b}{t}$ (cf. Line 12 of Algorithm 5). Therefore, at the $(t+1)$-th round, the $\boldsymbol{x}$-player would receive $-A\check{\boldsymbol{y}}_{t+1} = -A\boldsymbol{w}_t^*$

and compute

$$
\begin{aligned}
G^{\boldsymbol{w}}_{t+1} &= \langle \boldsymbol{w}_t, A^\top \boldsymbol{z}_t \rangle - \langle \check{\boldsymbol{y}}_{t+1}, A^\top \boldsymbol{z}_t \rangle \\
&= \langle \boldsymbol{w}_t - \boldsymbol{w}^*_t, A^\top \boldsymbol{z}_t \rangle > \frac{b}{t}
\end{aligned}
$$

which then enters the Line 4 of Algorithm 4.

Conversely, suppose that the $\boldsymbol{y}$-player does not enter Line 12 of Algorithm 5 at the t-th round (or, equivalently, plays OMD at the $(t+1)$-th round). Then $\langle \boldsymbol{w}_t - \boldsymbol{w}^*_t, A^\top \boldsymbol{z}_t \rangle \le \frac{b}{t}$, implying that

$$
\begin{aligned}
G^{\boldsymbol{w}}_{t+1} &= \langle \boldsymbol{w}_t - \boldsymbol{w}_{t+1}, A^\top \boldsymbol{z}_t \rangle \\
&\le \langle \boldsymbol{w}_t - \boldsymbol{w}^*_t, A^\top \boldsymbol{z}_t \rangle \le \frac{b}{t}
\end{aligned}
$$

hence preventing the $\boldsymbol{x}$-player from entering Line 4 of Algorithm 4.

Exactly the same computation holds when we reverse the role of $\boldsymbol{x}$- and $\boldsymbol{y}$-player. ■

Given Lemma 2.1, we now know that the $\boldsymbol{x}$-player switches to ROMD *if and only if* the $\boldsymbol{y}$-player does. The rest of the proof then readily follows from Theorems 2.2–2.3.

Theorem 2.4. *Suppose the $\boldsymbol{x}$-player plays according to Algorithm 4 for T rounds, and let R_T be the regret up to time T. Then*

1. *Let $T = T_1 + T_2 + T_3$ where T_1 is the number of OMD plays, T_2 is the number of ROMD plays, and T_3 is the number of signaling rounds (playing $\check{\boldsymbol{x}}_t$ or $\check{\boldsymbol{y}}_t$). Then there are constants C and C', depending only on m, n and $|A|_{\max}$, such that*

$$
\frac{1}{T}\mathrm{R}_T \le \frac{C \log T_1 + C'\sqrt{T_2}}{T_1 + T_2}. \tag{2.10}
$$

   In particular, if the opponent plays honestly, then $\mathrm{R}_T = O(\log T_1) = O(\log T)$. If the opponent is adversarial, we have $\mathrm{R}_T = O(\sqrt{T_2}) = O(\sqrt{T})$.

2. *Suppose that the honest $\boldsymbol{y}$-player plays Algorithm 5. Then the pair $(\boldsymbol{z}_T, \boldsymbol{w}_T)$ constitutes an $O\left(\frac{1}{T}\right)$-approximate equilibrium:*

$$
|V - \langle \boldsymbol{z}_T, A\boldsymbol{w}_T \rangle| \le \frac{C''}{T} \tag{2.11}
$$

   for some constant C''.

Proof. Suppose first that both players are honest.

We first prove the individual regret for the $\boldsymbol{x}$-player. We split the terms as follows:

$$\mathrm{R}_T = \mathrm{R}_{T_1}(\text{playing OMD}) + \mathrm{R}_{T_2}(\text{playing ROMD}) + \mathrm{R}_{T_3}(\text{signaling}). \tag{2.12}$$

Recall (2.6)-(2.9). We claim that

(a) $T_3 \le \lceil \log C_1 \rceil$.

(b) $T_2 \le 16 \cdot \frac{16^{T_3-1}-1}{15} := C_1'$.

Indeed, after $\lceil \log C_1 \rceil$-times signaling, we would have $b = 2^{T_3} > C_1$. Then (2.6) and (2.7) imply that we will never enter Line 12 again. On the other hand, we have

$$T_2 \le \sum_{r=1}^{T_3} 2^{4r} = \frac{16^{T_3-1}-1}{15}.$$

Combining (a), (b) and using (2.6), (2.8) in (2.12), we conclude that

$$\begin{aligned}\mathrm{R}_T &\le C_1 \log T_1 + C_2\sqrt{T_2} + 2|A|_{\max} T_3\\ &\le C_1 \log T_1 + C_2\sqrt{C_1'} + 2|A|_{\max}\lceil \log C_1 \rceil\\ &= O(\log T_1) = O(\log T)\end{aligned}$$

which establishes (2.10) in the honest case.

For convergence to the value of the game, we have, by (2.9),

$$|V - \langle \boldsymbol{z}_T, A\boldsymbol{w}_T\rangle| \le \frac{C_3}{T-T_2-T_3} \le \frac{C_3}{T-C^*}$$

where $C^* = \lceil \log C_1 \rceil + C_1'$. The proof of (2.11) is completed by using the fact that $\frac{1}{T-C^*} \le \frac{C^*}{T}$ when $T \ge \frac{C^{*2}}{C^*-1}$.

Finally, we show (2.10) in the adversarial case.

Let T_1, T_2, and T_3 be as before, and we again split the regret into:

$$\mathrm{R}_T = \mathrm{R}_{T_1}(\text{playing OMD}) + \mathrm{R}_{T_2}(\text{playing ROMD}) + \mathrm{R}_{T_3}(\text{signaling}).$$

Notice that this time the inequalities (2.6) and (2.7) do not apply since the opponent no longer plays OMD collaboratively. However, by Line 12 of Algorithm 4, for every OMD play we must have

$$\langle \boldsymbol{z}_t, -A\boldsymbol{w}_t\rangle - \left\langle \boldsymbol{z}_t^*, -A\boldsymbol{w}_t\right\rangle \le \frac{b}{t} \le \frac{2^{T_3}}{t}.$$

Following the analysis as in the honest setting, we may further write

$$\mathrm{R}_T \le 2^{T_3} \log T_1 + C_2\sqrt{T_2} + 2|A|_{\max} T_3.$$

It hence suffices to show that

$$2^{T_3} \log T_1 \le C^{**}\sqrt{T_1+T_2}. \tag{2.13}$$

for some constant C^{**}. To see (2.13), recall that

$$T_2 = \frac{16(16^{T_3}-1)}{15} \ge 16^{T_3-1}.$$

But then

$$\begin{aligned}\frac{2^{T_3}\log T_1}{\sqrt{T_1+T_2}} &\le \frac{2^{T_3}\log T_1}{\sqrt{2\sqrt{T_1T_2}}}\\ &\le \frac{2^{T_3}\log T_1}{2^{T_3-1}\cdot\sqrt{2}\cdot\sqrt[4]{T_1}} \le C^{**}\end{aligned}$$

for some universal constant C^{**}. ■

2.4 Experiments

The purpose of this section is to provide numerical evidence to the following claims of our theory:

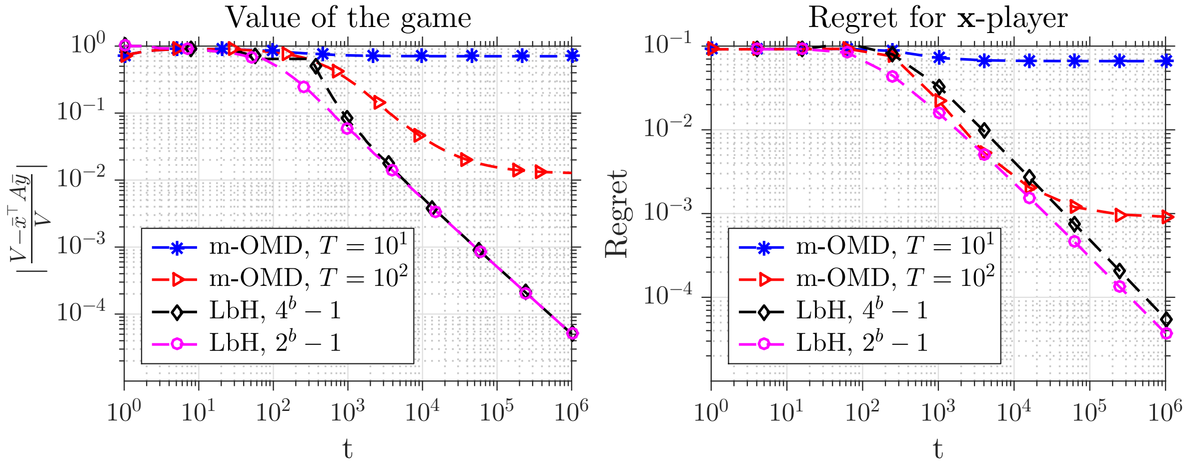
1. The LbH algorithm does not require knowing the time horizon beforehand, and our step-sizes are non-adaptive. Therefore, all quantities of interest, such as regrets or game value, should steadily decrease along the algorithm run.
2. The LbH algorithm automatically adjusts to honest and adversarial opponents.

For comparison, we include the modified OMD (henceforth abbreviated as m-OMD) of [RS13b] in our experiment, for different choices of time horizon.

We generate the entries of A uniformly at random in the interval $[-1,1]$, and we set $m=200$ and $n=300$.

We consider two scenarios:

1. *Honest setting*: Both players adhere to the prescribed algorithms and try to reach the Nash equilibrium collaboratively.
2. *Adversarial setting*: The $\boldsymbol{y}$-player greedily maximizes the instantaneous regret of the $\boldsymbol{x}$-player.



(a) Value of the game.

(b) Regret.

Figure 2.1: Honest setting.

2.4.1 Honest setting

The convergence for the honest setting is reported in Fig. 2.1, for two different parameter choices of LbH and m-OMD.

For both convergence to the game value and individual regret, after a short burn-in period (due to not knowing the C_1 in (2.6) and (2.7)), the LbH algorithm enters a steady $O\left(\frac{1}{T}\right)$-decreasing phase, as expected from our theory. On the other hand, as the m-OMD chooses step-sizes according to the time horizon, it eventually saturates in both plots.

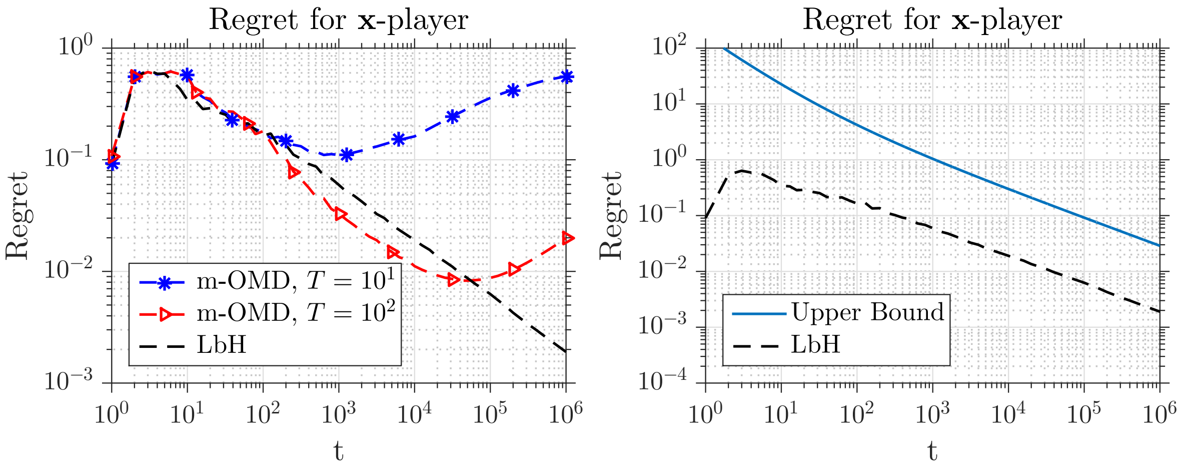
As noted by [RS13b], it is possible to prevent the saturation of m-OMD by employing the doubling trick or the techniques in [ACBG02]. However, doing so not only complicates the algorithm, but also introduces extra $\log T$ factors in the convergence of honest regret, since the doubling trick loses a $\log T$ factor for logarithmic regrets. Such rates are sub-optimal given our results.

2.4.2 Adversarial setting

We report the regret comparison in Fig. 2.2.

In the adversarial setting, the LbH algorithm is essentially running the ROMD, and hence we see a straight $O(T^{-\frac{1}{2}})$ decrease in the regret, as dictated by our upper bound in Theorem 2.3; see Fig. 2.2(b). The parameter choice does not effect the performance.

The m-OMD slightly outperforms LbH for a short period, but eventually blows up in regret. We remark that the short-term good empirical performance is due to the adaptive step-sizes of m-OMD, which require additional work per-iteration. Our LbH algorithm is non-adaptive, but is already competitive in terms of empirical performance.



(a) Regret comparison.

(b) Upper bound.

Figure 2.2: Adversarial setting.

3 Mirrored Langevin dynamics

In this chapter, we resolve the challenging task of provably sampling from *Latent Dirichlet Allocation* (LDA).

The difficulty of LDA stems from the fact that it is a *non-convex* sampling problem. However, by tapping into a connection between *Langevin dynamics* (LD) and MD (defined in Chapter 2), we discover that LDA in fact amounts to a simple *convex* sampling task, thus providing an elegant solution to a decades-old open problem.

3.1 Introduction

Many modern learning tasks involve sampling from a high-dimensional and large-scale distribution, which calls for algorithms that are scalable with respect to both the dimension and the data size. To this end, an powerful approach [WT11] is to discretize the *Langevin dynamics* (LD):

$$\mathrm{d}\boldsymbol{X}_t = -\nabla V(\boldsymbol{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\boldsymbol{B}_t, \tag{3.1}$$

where $e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}$ presents a target distribution and $\boldsymbol{B}_t$ is a d-dimensional Brownian motion. Such a framework has inspired numerous first-order sampling algorithms and has found wide empirical success [AKW12, CFG14, DFB$^+$14, DSM$^+$16, LS16, LZS16, PT13, SBCR16].

However, (3.1) is only known to converge when $e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}$ is log-concave (meaning V is convex) and *unconstrained.* On the other hand, many important distributions in ML are both non-log-concave *and* constrained. The focus of this chapter, the *Latent Dirichlet Allocation* (LDA), presents a typical and prominent example that is of paramount practical importance. Nonetheless, due to the reasons described above, there exists no provably convergent algorithm for *Latent Dirichlet Allocation* (LDA), in spite of the presence of several tailor-made schemes [LS16, PT13].

In this chapter, we resolve LDA by revealing that it is in fact a *log-concave* distribution in disguise. Concretely, our solution consists of three steps:

1. We first transform the constrained, non-log-concave LDA via the *entropic mirror map* to the *dual* distribution.

2. We show that the dual distribution of LDA is, surprisingly, unconstrained *and* log-concave!

3. Building upon a deep result in the theory of optimal transport [Vil08], we show that convergence in the primal and dual distributions are equivalent.

Combining 1-3, we readily prove that LD applied to the dual distribution of LDA is provably convergent. Finally, we demonstrate that the algorithm empirically outperforms the state-of-the-art.

3.2 Preliminaries

3.2.1 Notation

We use $\mathcal{C}^k$ to denote k-times differentiable functions with continuous k^{th} derivative. The Fenchel dual [Roc15] of a function h is denoted by $h^\star$. Given two mappings T, F of proper dimensions, we denote their composite map by $T \circ F$. For a probability measure μ, we write $\boldsymbol{X} \sim \mu$ to mean that "$\boldsymbol{X}$ is a random variable whose probability law is μ".

3.2.2 Push-forward and optimal transport

Let $\mathrm{d}\mu = e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}$ be a probability measure with support $\mathcal{X} := \mathrm{dom}(V) = \{\boldsymbol{x} \in \mathbb{R}^d \mid V(\boldsymbol{x}) < +\infty\}$, and h be a convex function on $\mathcal{X}$. We assume:

The function h is closed, proper, $h \in \mathcal{C}^2$, and $\nabla^2 h \succ 0$ on $\mathcal{X} \subset \mathbb{R}^d$. (A3.1)

All measures have finite second moments. (A3.2)

All measures vanish on sets with Hausdorff dimension [Man83] at most $d-1$. (A3.3)

The gradient map ∇h induces a new probability measure $\mathrm{d}\nu := e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y}$ through $\nu(E) = \mu\left(\nabla h^{-1}(E)\right)$ for every Borel set E on $\mathbb{R}^d$. We say that ν is the *push-forward measure* of μ under ∇h, and we denote it by $\nabla h \# \mu = \nu$. If $\boldsymbol{X} \sim \mu$ and $\boldsymbol{Y} \sim \nu$, we will sometimes abuse the notation by writing $\nabla h \# \boldsymbol{X} = \boldsymbol{Y}$ to mean $\nabla h \# \mu = \nu$.

If $\nabla h \# \mu = \nu$, the triplet (μ, ν, h) must satisfy the Monge-Ampère equation:

$$e^{-V} = e^{-W \circ \nabla h} \det \nabla^2 h. \tag{3.2}$$

Using $(\nabla h)^{-1} = \nabla h^\star$ and $\nabla^2 h \circ \nabla h^\star = \left(\nabla^2 h^\star\right)^{-1}$, we see that (3.2) is equivalent to

$$e^{-W} = e^{-V\circ\nabla h^\star} \det \nabla^2 h^\star \tag{3.3}$$

which implies $\nabla h^\star \# \nu = \mu$.

The 2-Wasserstein distance between μ_1 and μ_2 is defined by

$$\mathcal{W}_2^2(\mu_1, \mu_2) := \inf_{T: T\#\mu_1 = \mu_2} \int \|\boldsymbol{x} - T(\boldsymbol{x})\|^2 \mathrm{d}\mu_1(\boldsymbol{x}). \tag{3.4}$$

3.3 Mirrored Langevin dynamics

3.3.1 Motivation and algorithm

We begin by briefly recalling the *mirror descent* (MD) algorithm for optimization. In order to minimize a function over a bounded domain, say $\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$, MD uses a mirror map h to transform the primal variable $\boldsymbol{x}$ into the dual space $\boldsymbol{y} := \nabla h(\boldsymbol{x})$, and then performs gradient updates in the dual: $\boldsymbol{y}^+ = \boldsymbol{y} - \beta\nabla f(\boldsymbol{x})$ for some step-size β. The mirror map h is chosen to adapt to the geometry of the constraint $\mathcal{X}$, which can often lead to faster convergence [NY83] or, more pivotal to this work, an *unconstrained* optimization problem [BT03].

Inspired by the MD framework, we would like to use the mirror map idea to remove the constraint for sampling problems. Toward this end, we first establish a simple fact [Vil03]:

Theorem 3.1. *Let h satisfy Assumption* (A3.1). *Suppose that $\boldsymbol{X} \sim \mu$ and $\boldsymbol{Y} = \nabla h(\boldsymbol{X})$. Then $\boldsymbol{Y} \sim \nu := \nabla h \# \mu$ and $\nabla h^\star(\boldsymbol{Y}) \sim \mu$.*

Proof. For any Borel set E, we have $\nu(E) = \mathbb{P}(\boldsymbol{Y} \in E) = \mathbb{P}\left(\boldsymbol{X} \in \nabla h^{-1}(E)\right) = \mu\left(\nabla h^{-1}(E)\right)$. Since ∇h is one-to-one, $\boldsymbol{Y} = \nabla h(\boldsymbol{X})$ if and only if $\boldsymbol{X} = \nabla h^{-1}(\boldsymbol{Y}) = \nabla h^\star(\boldsymbol{Y})$. ■

In the context of sampling, Theorem 3.1 suggests the following simple procedure: For any target distribution $e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}$ with support $\mathcal{X}$, we choose a mirror map h on $\mathcal{X}$ satisfying Assumption (A3.1), and we consider the *dual distribution* associated with $e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}$ and h:

$$e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y} := \nabla h \# e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}. \tag{3.5}$$

Theorem 3.1 dictates that if we are able to draw a sample $\boldsymbol{Y}$ from $e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y}$, then $\nabla h^\star(\boldsymbol{Y})$ immediately gives a sample for the desired distribution $e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}$. Furthermore, suppose for the moment that $\mathrm{dom}(h^\star) = \mathbb{R}^d$, so that $e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y}$ is unconstrained. Then we can simply exploit the classical Langevin Dynamics (3.1) to efficiently take samples from $e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y}$.

The above reasoning leads us to set up the *mirrored Langevin dynamics* (MLD):

$$MLD \equiv \left\{ \begin{array}{l} \mathrm{d}\boldsymbol{Y}_t = -(\nabla W \circ \nabla h)(\boldsymbol{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\boldsymbol{B}_t \\ \boldsymbol{X}_t = \nabla h^\star(\boldsymbol{Y}_t) \end{array} \right. . \tag{3.6}$$

Notice that the stationary distribution of $\boldsymbol{Y}_t$ in MLD is $e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y}$, since $\mathrm{d}\boldsymbol{Y}_t$ is nothing but the Langevin Dynamics (3.1) with $\nabla V \leftarrow \nabla W$. As a result, we have $\boldsymbol{X}_t \to \boldsymbol{X}_\infty \sim e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}$.

Using (3.2), we can equivalently write the $\mathrm{d}\boldsymbol{Y}_t$ term in (3.6) as

$$\mathrm{d}\boldsymbol{Y}_t = -\nabla^2 h(\boldsymbol{X}_t)^{-1}\Big(\nabla V(\boldsymbol{X}_t) + \nabla \log\det \nabla^2 h(\boldsymbol{X}_t)\Big)\mathrm{d}t + \sqrt{2}\mathrm{d}\boldsymbol{B}_t.$$

In order to arrive at a practical algorithm, we then discretize the MLD, giving rise to the following equivalent iterations:

$$\boldsymbol{y}^{t+1} - \boldsymbol{y}^t = \left\{ \begin{array}{l} -\beta^t \nabla W(\boldsymbol{y}^t) + \sqrt{2\beta^t}\boldsymbol{\xi}^t \\ -\beta^t \nabla^2 h(\boldsymbol{x}^t)^{-1}\Big(\nabla V(\boldsymbol{x}^t) + \nabla \log\det \nabla^2 h(\boldsymbol{x}^t)\Big) + \sqrt{2\beta^t}\boldsymbol{\xi}^t \end{array} \right. \tag{3.7}$$

where in both cases $\boldsymbol{x}^{t+1} = \nabla h^\star(\boldsymbol{y}^{t+1})$, $\boldsymbol{\xi}^t$'s are i.i.d. standard Gaussian, and β^t's are step-sizes. The first formulation in (3.7) is useful when ∇W has a tractable form, while the second one can be computed using solely the information of V and h.

Next, we turn to the convergence of discretized MLD. Since $\mathrm{d}\boldsymbol{Y}_t$ in (3.6) is the classical Langevin Dynamics, and since we have assumed that W is unconstrained, it is typically not difficult to prove the convergence of $\boldsymbol{y}^t$ to $\boldsymbol{Y}_\infty \sim e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y}$. However, what we ultimately care about is the guarantee on the primal distribution $e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}$. The purpose of the next theorem is to fill the gap between primal and dual convergence.

We consider three most common metrics in evaluating approximate sampling schemes, namely the 2-Wasserstein distance $\mathcal{W}_2$, the total variation d_{TV}, and the relative entropy $D(\cdot\|\cdot)$.

Theorem 3.2 (Convergence in $\boldsymbol{y}^t$ implies convergence in $\boldsymbol{x}^t$)**.** *For any h satisfying Assumption* (A3.1)*, we have $d_{\mathrm{TV}}(\nabla h\#\mu_1, \nabla h\#\mu_2) = d_{\mathrm{TV}}(\mu_1, \mu_2)$ and $D(\nabla h\#\mu_1\|\nabla h\#\mu_2) = D(\mu_1\|\mu_2)$. In particular, we have $d_{\mathrm{TV}}(\boldsymbol{y}^t, \boldsymbol{Y}_\infty) = d_{\mathrm{TV}}(\boldsymbol{x}^t, \boldsymbol{X}_\infty)$ and $D(\boldsymbol{y}^t\|\boldsymbol{Y}_\infty) = D(\boldsymbol{x}^t\|\boldsymbol{X}_\infty)$ in* (3.7)*.*

If, furthermore, h is ρ-strongly convex: $\nabla^2 h \succeq \rho I$. Then $\mathcal{W}_2(\boldsymbol{x}^t, \boldsymbol{X}_\infty) \le \frac{1}{\rho}\mathcal{W}_2(\boldsymbol{y}^t, \boldsymbol{Y}_\infty)$.

Proof. See Appendix B.1. ■

3.3.2 Sampling algorithms on simplex and LDA

We apply the discretized MLD (3.7) to the task of sampling from distributions on the probability simplex $\Delta_d := \{\boldsymbol{x} \in \mathbb{R}^d \mid \sum_{i=1}^d x_i \le 1, x_i \ge 0\}$, which is instrumental in many fields of machine learning and statistics.

On a simplex, the most natural choice of h is the entropic mirror map [BT03], which is well-known to be 1-strongly convex:

$$h(\boldsymbol{x}) = \sum_{\ell=1}^{d} x_i \log x_\ell + \left(1 - \sum_{\ell=1}^{d} x_\ell\right) \log\left(1 - \sum_{\ell=1}^{d} x_\ell\right), \text{ where } 0\log 0 := 0. \tag{3.8}$$

In this case, the associated dual distribution can be computed explicitly.

Lemma 3.1 (Sampling on a simplex with entropic mirror map)**.** *Let $e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}$ be the target distribution on Δ_d, h be the entropic mirror map* (3.8)*, and $e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y} := \nabla h \# e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}$. Then the potential W of the push-forward measure admits the expression*

$$W(\boldsymbol{y}) = V \circ \nabla h^\star(\boldsymbol{y}) - \sum_{\ell=1}^{d} y_\ell + (d+1) h^\star(\boldsymbol{y}) \tag{3.9}$$

where $h^\star(\boldsymbol{y}) = \log\left(1 + \sum_{\ell=1}^{d} e^{y_\ell}\right)$ is the Fenchel dual of h, which is strictly convex and 1-Lipschitz gradient.

Proof. See Appendix B.2. ■

Crucially, we have $\mathrm{dom}(h^\star) = \mathbb{R}^d$, so that the Langevin Dynamics for $e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y}$ is *unconstrained.*

Based on Lemma 3.1, we now present the surprising case of the *non-log-concave* Dirichlet posteriors, a distribution of central importance in topic modeling [BNJ03], for which the dual distribution $e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y}$ becomes *strictly log-concave.* Sampling from the Dirichlet posteriors is the major building block for LDA.

▼ **Example 3.3.1** (Dirichlet Posteriors)**.** *Given parameters $\alpha_1, \alpha_2, \ldots, \alpha_{d+1} > 0$ and observations $n_1, n_2, \ldots, n_{d+1}$ where n_ℓ is the number of appearance of category ℓ, the probability density function of the Dirichlet posterior is*

$$p(\boldsymbol{x}) = \frac{1}{C} \prod_{\ell=1}^{d+1} x_\ell^{n_\ell + \alpha_\ell - 1}, \quad \boldsymbol{x} \in \mathrm{int}(\Delta_d) \tag{3.10}$$

where C is a normalizing constant and $x_{d+1} := 1 - \sum_{\ell=1}^{d} x_\ell$. The corresponding V is

$$V(\boldsymbol{x}) = -\log p(\boldsymbol{x}) = \log C - \sum_{\ell=1}^{d+1} (n_\ell + \alpha_\ell - 1) \log x_\ell, \quad \boldsymbol{x} \in \mathrm{int}(\Delta_d).$$

The interesting regime of the Dirichlet posterior is when it is ***sparse****, meaning the majority of the n_ℓ's are zero and a few n_k's are large, say of order $O(d)$. It is also common to set $\alpha_\ell < 1$ for all ℓ in practice. Evidently, V is neither convex nor concave in this case, and no existing non-asymptotic*

Algorithm 6: Stochastic Mirrored Langevin Dynamics (SMLD)

```
Require: Target distribution $e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}$ where $V = \sum_{i=1}^N V_i$, step-sizes $\beta^t$, batch-size $b$
Find $W_i$ such that $e^{-NW_i} \propto \nabla h \# e^{-NV_i}$ for all $i$.
for $t \leftarrow 0, 1, \cdots, T-1$ do
    Pick a mini-batch $B$ of size $b$ uniformly at random.
    Update $\boldsymbol{y}^{t+1} = \boldsymbol{y}^t - \frac{\beta^t N}{b} \sum_{i \in B} \nabla W_i(\boldsymbol{y}^t) + \sqrt{2\beta^t}\boldsymbol{\xi}^t$
    $\boldsymbol{x}^{t+1} = \nabla h^\star(\boldsymbol{y}^{t+1})$                # Update only when necessary.
end for
return $\boldsymbol{x}^T$
```

rate can be applied. However, plugging V into (3.9) *gives*

$$W(\boldsymbol{y}) = \log C - \sum_{\ell=1}^{d} (n_\ell + \alpha_\ell) y_\ell + \left(\sum_{\ell=1}^{d+1} (n_\ell + \alpha_\ell) \right) h^\star(\boldsymbol{y}) \tag{3.11}$$

which, magically, becomes strictly convex and $O(d)$-Lipschitz gradient ***no matter what the observations and parameters are!*** *In view of Theorem 3.2 and [DMM18,* **Corollary 7***], one can then apply* (3.7) *to obtain an $\tilde{O}\left(\epsilon^{-2} d^2 R_0\right)$ convergence in relative entropy, where $R_0 :=$ $\mathcal{W}_2^2(\boldsymbol{y}^0, e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y})$ is the initial Wasserstein distance to the target.* ■

3.4 Stochastic mirrored Langevin dynamics

We have thus far only considered deterministic methods based on exact gradients. In practice, however, evaluating gradients typically involves one pass over the full data, which can be time-consuming in large-scale applications. In this section, we turn attention to the *mini-batch* setting, where one can use a small subset of data to form stochastic gradients.

Toward this end, we assume:

$$\text{The distribution } e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x} \text{ admits a decomposable structure } V = \sum_{i=1}^{N} V_i. \tag{3.12}$$

Consider the following common scheme in obtaining stochastic gradients. Given a batch-size b, we randomly pick a mini-batch B from $\{1, 2, \ldots, N\}$ with $|B| = b$, and form an unbiased estimate of ∇V by computing

$$\tilde{\nabla} V := \frac{N}{b} \sum_{i \in B} \nabla V_i. \tag{3.13}$$

The following lemma asserts that exactly the same procedure can be carried out in the dual.

Lemma 3.2. *Assume that h is 1-strongly convex. For $i = 1, 2, ..., N$, let W_i be such that*

$$e^{-NW_i} = \nabla h \# \frac{e^{-NV_i}}{\int e^{-NV_i}}. \tag{3.14}$$

Define $W := \sum_{i=1}^N W_i$ *and* $\tilde{\nabla} W := \frac{N}{b}\sum_{i\in B} \nabla W_i$, *where* B *is chosen as in* (3.13)*. Then:*

1. *Primal decomposibility implies dual decomposability: There is a constant* C *such that* $e^{-(W+C)} = \nabla h \# e^{-V}$.
2. *For each* i, *the gradient* ∇W_i *depends only on* ∇V_i *and the mirror map* h.
3. *The gradient estimate is unbiased:* $\mathbb{E}\tilde{\nabla} W = \nabla W$.
4. *The dual stochastic gradient is more accurate:* $\mathbb{E}\|\tilde{\nabla} W - \nabla W\|^2 \le \mathbb{E}\|\tilde{\nabla} V - \nabla V\|^2$.

Proof. See Appendix B.3. ■

Lemma 3.2 furnishes a template for the mini-batch extension of MLD. The pseudocode is detailed in Algorithm 6, whose convergence rate is given by the next theorem.

Theorem 3.3. *Let* $e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}$ *be a distribution satisfying Assumption* (3.12)*, and* h *a 1-strongly convex mirror map. Let* $\sigma^2 := \mathbb{E}\|\tilde{\nabla} V - \nabla V\|^2$ *be the variance of the stochastic gradient of* V *in* (3.13)*. Suppose that the corresponding dual distribution* $e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y} = \nabla h \# e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}$ *satisfies* $LI \succeq \nabla^2 W \succeq 0$. *Then, applying SMLD with constant step-size* $\beta^t = \beta$ *yields*[1]*:*

$$D\left(\boldsymbol{x}^T \| e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}\right) \le \sqrt{\frac{2\mathcal{W}_2^2\left(\boldsymbol{y}^0, e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y}\right)\left(Ld+\sigma^2\right)}{T}} = O\left(\sqrt{\frac{Ld+\sigma^2}{T}}\right), \tag{3.15}$$

provided that $\beta \le \min\left\{\left[2T\mathcal{W}_2^2\left(\boldsymbol{y}^0, e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y}\right)\left(Ld+\sigma^2\right)\right]^{-\frac{1}{2}}, \frac{1}{L}\right\}$.

Proof. See Appendix B.4. ■

▼ **Example 3.4.1** (SMLD for Dirichlet Posteriors)**.** *For the case of Dirichlet posteriors, we have seen in* (3.11) *that the corresponding dual distribution satisfies* $(N+\Gamma)I \succeq \nabla^2 W \succ 0$, *where* $N := \sum_{\ell=1}^{d+1} n_\ell$ *and* $\Gamma := \sum_{\ell=1}^{d+1} \alpha_\ell$. *Furthermore, it is easy to see that the stochastic gradient* $\tilde{\nabla} W$ *can be efficiently computed (see Appendix B.5):*

$$\tilde{\nabla} W(\boldsymbol{y})_\ell := \frac{N}{b}\sum_{i\in B} \nabla W_i(\boldsymbol{y})_\ell = -\left(\frac{Nm_\ell}{b} + \alpha_\ell\right) + (N+\Gamma)\frac{e^{y_\ell}}{1+\sum_{k=1}^d e^{y_k}}, \tag{3.16}$$

where m_ℓ *is the number of observations of category* ℓ *in the mini-batch* B. *As a result, Theorem 3.3 states that SMLD achieves*

$$D\left(\boldsymbol{x}^T \| e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}\right) \le \sqrt{\frac{2\mathcal{W}_2^2\left(\boldsymbol{y}^0, e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y}\right)\left((N+\Gamma)(d+1)+\sigma^2\right)}{T}} = O\left(\sqrt{\frac{(N+\Gamma)d+\sigma^2}{T}}\right)$$

---

[1] Our guarantee is given on a randomly chosen iterate from $\{\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^T\}$, instead of the final iterate $\boldsymbol{x}^T$. In practice, we observe that the final iterate always gives the best performance, and we will ignore this minor difference in the theorem statement.

with a constant step-size. ■

3.5 Experiments

We conduct experiments with a two-fold purpose. First, we use a low-dimensional synthetic data, where we can evaluate the total variation error by comparing histograms, to verify the convergence rates in our theory. Second, We demonstrate that the SMLD, modulo a necessary modification for resolving numerical issues, outperforms state-of-the-art first-order methods on the *Latent Dirichlet Allocation* (LDA) application with Wikipedia corpus.

3.5.1 Synthetic experiment for Dirichlet posterior

We implement the deterministic MLD for sampling from an 11-dimensional Dirichlet posterior (3.10) with $n_1 = 10000$, $n_2 = n_3 = 10$, and $n_4 = n_5 = \cdots = n_{11} = 0$, which aims to capture the sparse nature of real observations in topic modeling. We set $\alpha_\ell = 0.1$ for all ℓ.

As a baseline comparison, we include the *Stochastic Gradient Riemannian Langevin Dynamics* (SGRLD) [PT13] with the expanded-mean parametrization. SGRLD is a tailor-made first-order scheme for simplex constraints, and it remains one of the state-of-the-art algorithms for LDA. For fair comparison, we use deterministic gradients for SGRLD.

We perform a grid search over the constant step-size for both algorithms, and we keep the best three for MLD and SGRLD.

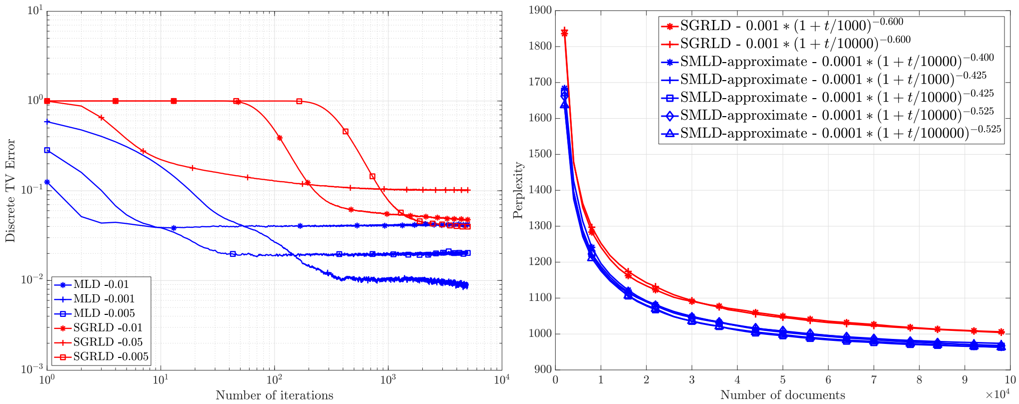
Fig. 3.1(a) reports the total variation error along the first dimension, where we can see that MLD outperforms SGRLD by a substantial margin. As dictated by our theory, all the MLD curves decay at the $O(T^{-\frac{1}{2}})$ rate until they saturate at the dicretization error level. In contrast, SGRLD lacks non-asymptotic guarantees, and there is no clear convergence rate we can infer from Fig. 3.1(a).

The improvement along all other dimensions (i.e., topics with less observations) are even more significant; see Appendix B.6.1.

3.5.2 Latent Dirichlet Allocation with Wikipedia corpus

An influential framework for topic modeling is the *Latent Dirichlet Allocation* (LDA) [BNJ03], which, given a text collection, requires to infer the posterior word distributions without knowing the exact topic for each word. The full model description is standard but somewhat convoluted; we refer to the classic [BNJ03] for details.

Each topic k in LDA determines a word distribution $\boldsymbol{\pi}_k$, and suppose there are in total K topics and $W+1$ words. The variable of interest is therefore $\boldsymbol{\pi} := (\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_K) \in \Delta_W \times \Delta_W \times \cdots \Delta_W$. Since this domain is a Cartesian product of simplices, we propose to use $\tilde{h}(\boldsymbol{\pi}) := \sum_{k=1}^K h(\boldsymbol{\pi}_k)$,



(a) Synthetic data, first dimension.

(b) LDA on Wikipedia corpus.

where h is the entropic mirror map (3.8), for SMLD. It is easy to see that all of our computations for Dirichlet posteriors generalize to this setting.

Experimental setup

We implement the SMLD for LDA on the Wikipedia corpus with 100000 documents, and we compare the performance against the SGRLD [PT13]. In order to keep the comparison fair, we adopt exactly the same setting as in [PT13], including the model parameters, the batch-size, the Gibbs sampler steps, etc. See Section 4 and 5 in [PT13] for omitted details.

Another state-of-the-art first-order algorithm for LDA is the SGRHMC in [MCF15], for which we skip the implementation, due to not knowing how the $\hat{B}_t$ was chosen in [MCF15]. Instead, we will repeat the same experimental setting as [MCF15] and directly compare our results versus the ones reported in [MCF15]. See Appendix B.6.2 for comparison against SGRHMC.

A numerical trick and the SMLD-approximate algorithm

A major drawback of the SMLD in practice is that the stochastic gradients (3.16) involve exponential functions, which are unstable for large-scale problems. For instance, in python, `np.exp(800) = inf`, whereas the relevant variable regime in this experiment extends to 1600. To resolve such numerical issues, we appeal to the linear approximation[2] $\exp(\boldsymbol{y}) \simeq \max\{0, 1+\boldsymbol{y}\}$. Admittedly, our theory no longer holds under such numerical tricks, and we shall not claim that our algorithm is provably convergent for LDA. Instead, the contribution of MLD here is to identify the dual dynamics associated with (3.11), which would have been otherwise difficult to perceive. We name the resulting algorithm "SMLD-approximate" to indicate its heuristic nature.

---

[2] One can also use a higher-order Taylor approximation for $\exp(\boldsymbol{y})$, or add a small threshold $\exp(\boldsymbol{y}) \simeq \max\{\epsilon, 1+\boldsymbol{y}\}$ to prevent the iterates from going to the boundary. In practice, we observe that these variants do not make a huge impact on the performance.

Results

Fig. 3.1(b) reports the perplexity on the test data up to 100000 documents, with the five best step-sizes we found via grid search for SMLD-approximate. For SGRLD, we use the best step-sizes reported in [PT13].

From the figure, we can see a clear improvement, both in terms of convergence speed and the saturation level, of the SMLD-approximate over SGRLD. One plausible explanation for such phenomenon is that our MLD, as a simple unconstrained Langevin Dynamics, is less sensitive to discretization. On the other hand, the underlying dynamics for SGRLD is a more sophisticated Riemannian diffusion, which requires finer discretization than MLD to achieve the same level of approximation to the original continuous-time dynamics, and this is true even in the presence of noisy gradients and our numerical heuristics

4 Spurious convergence of min-max optimization algorithms

Non-convex/non-concave min-max optimization is a subject of intensive study. However, in this chapter, we rigorously establish the negative result that most algorithms proposed for solving non-convex/non-concave **Task 3** exhibit problematic asymptotic behavior and cannot serve as general solutions.

4.1 Introduction

Consider a min-max optimization – or *saddle-point* – problem of the form

$$\min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{y}\in\mathcal{Y}} F(\boldsymbol{x},\boldsymbol{y}) \tag{SP}$$

where $\mathcal{X}$, $\mathcal{Y}$ are subsets of a Euclidean space and $F\colon \mathcal{X}\times\mathcal{Y}\to\mathbb{R}$ may be non-convex/non-concave. Given an algorithm for solving (SP), the following fundamental questions arise:

When does the algorithm converge? Where does the algorithm converge to? (⋆)

The goal of this chapter is to provide concrete answers to (⋆) and to study their practical implications for a wide array of existing methods.

Min-max problems of this type have found widespread applications in machine learning in the context of GANs [GPAM$^+$14], robust reinforcement learning [PDSG17], and other models of adversarial training [MMS$^+$18]. In this broad setting, it has become empirically clear that the joint training of two neural networks (NNs) with competing objectives is fundamentally more difficult than training a *single* NN of similar size and architecture. The latter task boils down to successfully finding a (good) local minimum of a non-convex function, so it is instructive to revisit (⋆) in the context of (non-convex) minimization problems.

In this case, much of the theory on stochastic gradient descent (SGD) methods – the "gold standard" for deep NN training – can be informally summed up as follows:

1. Bounded trajectories of SGD always converge to a set of critical points [Lju77, Lju86, BT00].
2. The limits of SGD do not contain saddle points or other spurious solutions [Pem90, BD96, GHJY15].

At first glance, these positive results might raise high expectations for solving (SP). Unfortunately, one can easily find counterexamples with very simple *bilinear* games of the form $F(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top A \boldsymbol{y}$: naïvely applying stochastic gradient descent/ascent (SGDA) methods in this case leads to recurrent orbits that do not contain *any* critical point of F. Such a phenomenon has no counterpart in non-convex minimization, and is fundamentally tied to the min-max structure of (SP).

The failure of SGDA in bilinear games has been studied extensively [YSX$^+$18, GBV$^+$19, ALW19, AMLJG19, GHP$^+$19, MOP19, MPP18, LS19, SA19, ZY19, PDZC20], leading to more sophisticated schemes such as stochastic extra-gradient (SEG) methods and their variants [DISZ18, MLZ$^+$19, GBV$^+$19, HIMM19, CGFLJ19]. Meanwhile, to bypass such globally oscillatory issues, another thread of research [HRU$^+$17, NK17, DP18, MLZ$^+$19, ADLH19, MJS19, NSH$^+$19, JNJ19, LMR$^+$19, RCJ19, MRS20] has shifted its attention to *local analysis.* Essentially, these works either analyze the algorithmic behaviors only "sufficiently close" to critical points, or impose stringent assumptions on F (such as "coherence" [MLZ$^+$19] or the existence of solutions to a Minty variational inequality [LMR$^+$19]) to ensure the equivalence between global and local convergence.

Although these studies have certainly led to fruitful results, the realm beyond bilinear games and (locally) idealized objectives remains somewhat unexplored (with a few exceptions that we discuss in detail below). In particular, a convergence theory for general non-convex/non-concave problems is still lacking.

Our contributions. In this chapter, we aim to bridge this gap by providing precise answers to ($\star$) for a wide range of min-max optimization algorithms that can be seen as *generalized Robbins–Monro (RM) schemes* [RM51]. Mirrorring the minimization perspective, we prove that, for any such algorithm $\mathcal{A}$:

1. Bounded trajectories of $\mathcal{A}$ always converge to an *internally chain-transitive* (ICT) set.
2. Trajectories of $\mathcal{A}$ may converge with arbitrarily high probability to spurious attractors that contain *no* critical point of F.

The most critical implication of our theory is that one can reduce the long-term behavior of a training algorithm to its associated ICT sets, a notion deeply rooted in the study of dynamical systems [Bow75, Con78, BH96, Ben99, BHS05] that formalizes the idea of "discrete limits of continuous flows"; cf. Section 4.4. As an example, in minimization problems, one can prove that the ICT sets of SGD consist solely of components of critical points; on the other hand, we show that ICT sets in min-max optimization can exhibit drastically more complicated structures, even when $\mathcal{X} = \mathcal{Y} = \mathbb{R}$. In particular, we establish the following negative results:

- An ICT set may contain (almost) *globally attracting limit cycles*, and the algorithms designed to eliminate periodic orbits in bilinear games *cannot escape them.* This observation corroborates the persistence of non-convergent behaviors in GAN training, and suggests that bilinear games may be insufficient as models for such applications.

- There exist *unstable* critical points whose neighborhood contains an (almost) *globally stable* ICT set. Therefore, in sharp contrast to minimization problems, "avoiding unstable critical points" *does not imply* "escaping unstable critical points" in min-max problems.

- There exist *stable* min-max points whose basin of attraction is "shielded" by an *unstable* ICT set. As a result, existing algorithms are repelled from a desirable solution with high probability, even if initialized arbitrarily close to it.

Finally, we provide numerical illustrations of the above, which further show that common practical tweaks (such as averaging or adaptive algorithms) also fail to address these problematic cases.

Further related work. To our knowledge, the convergence to non-critical sets in (SP) has only been systematically studied in a few settings. Besides the bilinear games alluded to above, other instances include the "almost bilinear games" [ALW19] and deterministic gradient descent/ascent (GDA) applied to "hidden bilinear games" [FVGP19]. In contrast to these works, our framework does not impose any structural assumption and requires only mild regularity of F, and our results apply to many existing methods beyond (S)GDA; cf. Section 4.3. The generality of our approach is made possible by foundational results in dynamical systems [Ben99, BH96], which have not been exploited before in the context of min-max optimization, and have only recently been applied to learning in games with the aim of showing convergence to (local) Nash equilibria [PL12, BHS05, BHS06, PML17, MRS20, CHM17, BM17, BLM18, MZ19, BBF18].

Another work [Let20] has a similar motivation to our study. The focus of [Let20] is on providing counterexamples that rule out the convergence of deterministic "reasonable" and "global" algorithms. There are two major distinctions that make our approaches complementary: [Let20] focuses on the impossibility of *desirable* convergence guarantees in a purely *deterministc* setting; in contrast, our paper focuses squarely on the occurrence of *undesirable convergence* phenomena with probability 1 in *stochastic* algorithms. Taken together, the work [Let20] and our own paint a fairly complete picture of the fundamental limits of min-max optimization algorithms.

4.2 Setup and preliminaries

We focus on general problems of the form (SP) with $\mathcal{X} = \mathbb{R}^{d_\mathcal{X}}$, $\mathcal{Y} = \mathbb{R}^{d_\mathcal{Y}}$, and F assumed C^1. To ease notation, we will denote $z = (\boldsymbol{x}, \boldsymbol{y})$, $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and $d = d_\mathcal{X} + d_\mathcal{Y}$. In addition, we will write

$$V(z) \equiv (V_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{y}), V_{\boldsymbol{y}}(\boldsymbol{x}, \boldsymbol{y})) := (-\nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{y}), \nabla_{\boldsymbol{y}} F(\boldsymbol{x}, \boldsymbol{y})) \tag{4.1}$$

for the (min-max) gradient field of F, and we will assume that V is Lipschitz. In some cases we will also require V to be C^1 and we will write $J(z)$ for its Jacobian; this additional assumption will be stated explicitly whenever invoked.

A *solution* of (SP) is a tuple $z^\star = (\boldsymbol{x}^\star, \boldsymbol{y}^\star)$ with $F(\boldsymbol{x}^\star, \boldsymbol{y}) \leq F(\boldsymbol{x}^\star, \boldsymbol{y}^\star) \leq F(\boldsymbol{x}, \boldsymbol{y}^\star)$ for all $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{y} \in \mathcal{Y}$; likewise, a *local solution* of (SP) is a tuple $(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$ that satisfies this inequality locally. Finally, a state $z^\star$ with $V(z^\star) = 0$ is said to be a *critical* (or *stationary*) *point* of F. When V is C^1, any local solution is a *stable* critical point [JNJ19], i.e., $\nabla^2_{\boldsymbol{x}} F(\boldsymbol{x}^\star, \boldsymbol{y}^\star) \succeq 0$ and $\nabla^2_{\boldsymbol{y}} F(\boldsymbol{x}^\star, \boldsymbol{y}^\star) \preceq 0$.

From an algorithmic standpoint, we will focus exclusively on the black-box optimization paradigm [Nes04] with *stochastic first-order oracle* (SFO) feedback; algorithms with a more complicated feedback structure (such as a best-response oracle [JNJ19, NI19, FCR19]) or based on mixed-strategy sampling [HLC19, DEJM$^+$20] are not considered in this chapter. In detail, when called at $z = (\boldsymbol{x}, \boldsymbol{y})$ with random seed $\omega \in \Omega$, an SFO returns a random vector $\mathsf{V}(z;\omega) \equiv (\mathsf{V}_{\boldsymbol{x}}(z;\omega), \mathsf{V}_{\boldsymbol{y}}(z;\omega))$ of the form

$$\mathsf{V}(z;\omega) = V(z) + \mathsf{U}(z;\omega) \tag{SFO}$$

where the error term $\mathsf{U}(z;\omega)$ captures all sources of uncertainty in the model (e.g., the selection of a minibatch in GAN training models, system state observations in reinforcement learning, etc.). Regarding this error term, we will assume throughout that it is zero-mean and sub-Gaussian:

$$\mathbb{E}[\mathsf{U}(z;\omega)] = 0 \quad \text{and} \quad \mathbb{P}(\|\mathsf{U}(z;\omega)\| \geq t) \leq 2e^{-\frac{t^2}{2\sigma^2}} \tag{4.2}$$

for some $\sigma > 0$ and all $z \in \mathcal{Z}$. The sub-Gaussian tail assumption is standard in the literature [Nes04, Nes09, NJLS09, JNT11], and it can be further relaxed with little loss of generality to finite variance $\mathbb{E}[\|\mathsf{U}(z;\omega)\|^2] \leq \sigma^2$. To streamline our discussion, we will present our results in the sub-Gaussian regime and we will rely on a series of remarks to explain any modifications required for different assumptions on U.

4.3 Core algorithmic framework

4.3.1 The Robbins–Monro template

Much of our analysis will focus on iterative algorithms that can be cast in the abstract Robbins–Monro framework of stochastic approximation [RM51]:

$$Z_{n+1} = Z_n + \gamma_n [V(Z_n) + W_n] \tag{RM}$$

where:

1. $Z_n = (X_n, Y_n) \in \mathcal{Z}$ denotes the state of the algorithm at each stage $n = 1, 2, \dots$
2. W_n is a generalized error term (described in detail below).
3. γ_n is the step-size (a hyperparameter, typically of the form $\gamma_n \propto 1/n^p$, $p \geq 0$).

In the above, the error term W_n is generated *after* Z_n; thus, by default, W_n is not adapted to the history (natural filtration) $\mathcal{F}_n := \mathcal{H}(Z_1, \dots, Z_n)$ of Z_n. For concision, we will write

$$V_n = V(Z_n) + W_n \tag{4.3}$$

so V_n can be seen as a noisy estimate of $V(Z_n)$. In more detail, to differentiate between "random" (zero-mean) and "systematic" (non-zero-mean) errors in V_n, it will be convenient to further decompose the error process W_n as

$$W_n = U_n + b_n \tag{4.4}$$

where $b_n = \mathbb{E}[W_n \,|\, \mathcal{F}_n]$ represents the systematic component of the error and $U_n = W_n - b_n$ captures the random, zero-mean part. In view of all this, we will consider the following descriptors for W_n:

$$a)\quad \textit{Bias:}\quad B_n = \|b_n\| \tag{4.5a}$$

$$b)\quad \textit{Variance:}\quad \sigma_n^2 = \mathbb{E}[\|U_n\|^2] \tag{4.5b}$$

The precise behavior of B_n and σ_n^2 will be examined on a case-by-case basis below.

4.3.2 Specific algorithms

In the rest of this section, we discuss how a wide range of algorithms used in the literature can be seen as special instances of the general template (RM) above.

▼ **Algorithm 4.1** (Stochastic gradient descent/ascent)**.** The basic SGDA algorithm – also known as the *Arrow–Hurwicz* method [AHU58] – queries an SFO and proceeds as:

$$Z_{n+1} = Z_n + \gamma_n \mathsf{V}(Z_n; \omega_n), \tag{SGDA}$$

where $\omega_n \in \Omega$ $(n = 1, 2, \dots)$ is an independent and identically distributed (i.i.d.) sequence of oracle seeds. As such, (SGDA) admits a straightforward RM representation by taking $W_n = U_n = \mathsf{U}(Z_n; \omega_n)$ and $b_n = 0$. ▲

▼ **Algorithm 4.2** (Alternating stochastic gradient descent/ascent)**.** A common variant of SGDA, is to *alternate* the updates of the min/max variables, resulting in the *alternating stochastic gradient descent/ascent* (alt-SGDA) method:

$$\begin{aligned} X_{n+1} &= X_n + \gamma_n \mathsf{V}_{\boldsymbol{x}}(X_n, Y_n; \omega_n) &&= X_n + \gamma_n [V_{\boldsymbol{x}}(X_n, Y_n) + U_{\boldsymbol{x},n}] \\ Y_{n+1} &= Y_n + \gamma_n \mathsf{V}_{\boldsymbol{y}}(X_{n+1}, Y_n; \omega_n^+) &&= Y_n + \gamma_n [V_{\boldsymbol{y}}(X_{n+1}, Y_n) + U_{\boldsymbol{y},n}] \end{aligned} \tag{alt-SGDA}$$

where ω_n, ω_n^+ $(n = 1, 2, \dots)$ are sequences of i.i.d. random seeds, $U_{\boldsymbol{x},n} \coloneqq \mathsf{U}_{\boldsymbol{x}}(X_n, Y_n; \omega_n)$, and $U_{\boldsymbol{y},n} \coloneqq \mathsf{U}_{\boldsymbol{y}}(X_{n+1}, Y_n; \omega_n^+)$. The RM representation of (alt-SGDA) is obtained by taking $Z_n = (X_n, Y_n)$, $b_n = (0, V_{\boldsymbol{y}}(X_{n+1}, Y_n) - V_{\boldsymbol{y}}(X_n, Y_n))$, and $U_n = (U_{\boldsymbol{x},n}, U_{\boldsymbol{y},n})$. ▲

▼ **Algorithm 4.3** (Stochastic extra-gradient)**.** Going beyond (SGDA), the (stochastic) extra-gradient algorithm exploits the following principle [Kor76, Nem04, JNT11]: given a “base” state Z_n, the algorithm queries the oracle at Z_n to generate a *leading* state Z_n^+ and then updates Z_n with oracle information from Z_n^+. Assuming SFO feedback as above, this process may be described as follows:

$$\begin{aligned} Z_n^+ &= Z_n + \gamma_n \mathsf{V}(Z_n; \omega_n), \\ Z_{n+1} &= Z_n + \gamma_n \mathsf{V}(Z_n^+; \omega_n^+). \end{aligned} \tag{SEG}$$

To recast (SEG) in the Robbins–Monro framework, simply take $W_n = \mathsf{V}(Z_n^+; \omega_n^+) - V(Z_n)$, i.e., $U_n = \mathsf{U}(Z_n^+; \omega_n^+)$ and $b_n = V(Z_n^+) - V(Z_n)$. ▲

▼ **Algorithm 4.4** (Optimistic gradient / Popov’s extra-gradient)**.** Compared to (SGDA), the scheme (SEG) involves two oracle queries per iteration, which is considerably more costly. An alternative iterative method with a single oracle query per iteration was proposed by [Pop80]:

$$\begin{aligned} Z_n^+ &= Z_n + \gamma_n \mathsf{V}(Z_{n-1}^+; \omega_{n-1}), \\ Z_{n+1} &= Z_n + \gamma_n \mathsf{V}(Z_n^+; \omega_n). \end{aligned} \tag{OG/PEG}$$

Its Robbins–Monro representation is obtained by setting $W_n = \mathsf{V}(Z_n^+; \omega_n) - V(Z_n)$, i.e., $U_n = \mathsf{U}(Z_n^+; \omega_n)$ and $b_n = V(Z_n^+) - V(Z_n)$.

Popov’s extra-gradient has been rediscovered several times and is more widely known as the optimistic gradient (OG) method in the machine learning literature [RS13a, CYL⁺12, DISZ18, HIMM19]. In unconstrained min-max optimization, (OG/PEG) turns out to be equivalent to a number of other existing methods, including “extrapolation from the past” [GBV⁺19], reflected gradient [MT20], and the “prediction method” of [YSX⁺18]. ▲

▼ **Algorithm 4.5** (Kiefer–Wolfowitz)**.** When first-order feedback is unavailable, a popular alternative is to obtain gradient information of F via zeroth-order observations [LLC^{+}19]. This idea can be traced back to the seminal work of [KW52] and the subsequent development of the *simultaneous perturbation stochastic approximation* (SPSA) method by [Spa92]. In our setting, this leads to the recursion:

$$\begin{aligned} V_n &= \pm(d/\delta_n)\, F(Z_n + \delta_n \omega_n)\, \omega_n \\ Z_{n+1} &= Z_n + \gamma_n V_n \end{aligned} \tag{SPSA}$$

where $\delta_n \searrow 0$ is a vanishing "sampling radius" parameter, ω_n is drawn uniformly at random from the composite basis $\Omega = \mathcal{E}_{\mathcal{X}} \cup \mathcal{E}_{\mathcal{Y}}$ of $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, and the "±" sign is equal to -1 if $\omega_n \in \mathcal{E}_{\mathcal{X}}$ and $+1$ if $\omega_n \in \mathcal{E}_{\mathcal{Y}}$. Viewed this way, the interpretation of (SPSA) as a Robbins–Monro method is immediate; furthermore, a straightforward calculation (that we defer to the appendix) shows that the sequence of gradient estimators V_n in (SPSA) has $B_n = \mathcal{O}(\delta_n)$ and $\sigma_n^2 = \mathcal{O}(1/\delta_n^2)$. ▲

Further examples that can be cast in the general framework (RM) include the negative momentum method [GHP^{+}19], generalized OG schemes [MOP19], and centripetal acceleration [PDZC20]; the analysis is similar and we omit the details. Certain scalable second-order methods can also be viewed as Robbins–Monro schemes, but the driving vector field V is no longer the gradient field of F; we discuss this in Example 4.5.3 and the appendix.

4.4 Convergence analysis

4.4.1 Continuous vs. discrete time

The main idea of our approach will be to treat (RM) as a noisy discretization of the *mean dynamics*

$$\dot{z}(t) = V(z(t)). \tag{MD}$$

This is motivated by the fact that $\dot{z}(t)$ can be seen as the continuous-time limit of the finite difference quotient $(Z_{n+1} - Z_n)/\gamma_n$: in this way, if the error term W_n in (RM) is sufficiently well-behaved, it is plausible to expect that the iterates of (RM) and the solutions of (MD) eventually come together. This approach has proved very fruitful when the mean dynamics (MD) comprise a *gradient system*, i.e., $V = -\nabla f$ for some (possibly non-convex) $f\colon \mathcal{Z} \to \mathbb{R}$. In this case (and modulo mild assumptions), the systems (RM) and (MD) both converge to the critical set of f, see e.g., [Lju77, KC78, BMP90, KY97, BT00].

On the other hand, the min-max landscape is considerably more involved. The most widely known illustration is given by the bilinear objective $F(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}\boldsymbol{y}$: in this case (see Fig. 4.1), the trajectories (MD) comprise periodic orbits of perfect circles centered at the origin (the unique critical point of F). However, the behavior of different RM schemes can vary wildly, even in the absence of noise ($\sigma = 0$): trajectories of (SGDA) spiral outwards, each converging

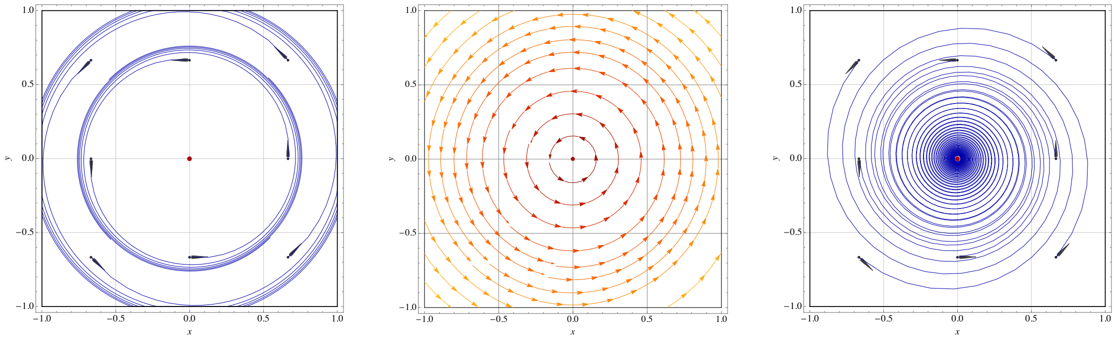


Figure 4.1: Comparison of different RM schemes for bilinear games $F(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}\boldsymbol{y}$, $\boldsymbol{x},\boldsymbol{y} \in \mathbb{R}$. From left to right: (*a*) gradient descent/ascent; (*b*) the mean dynamics (MD); (*c*) extra-gradient.

to an (initialization-dependent) periodic orbit; instead, trajectories of (SEG) spiral inwards, eventually converging to the solution $z^\star = (0,0)$.

This particular difference between gradient and extra-gradient schemes has been well-documented in the literature, cf. [DISZ18, GBV$^+$19, MLZ$^+$19]. More pertinent to our theory, it also raises several key questions:

1. *What is the precise link between RM methods and the mean dynamics* (MD)*?*
2. *When can* (MD) *accurately predict the long-run behavior of an RM method?*

The rest of this section is devoted to providing precise answers to these questions.

4.4.2 Stochastic approximation

We begin by introducing a measure of "closeness" between the iterates of (RM) and the solution orbits of (MD). To do so, let $\tau_n = \sum_{k=1}^{n} \gamma_k$ denote the "effective time" that has elapsed at the n-th iteration of (RM), and define the continuous-time interpolation $Z(t)$ of Z_n as

$$Z(t) = Z_n + \frac{t-\tau_n}{\tau_{n+1}-\tau_n}(Z_{n+1}-Z_n) \tag{4.6}$$

for all $t \in [\tau_n, \tau_{n+1}]$, $n \geq 1$. To compare $Z(t)$ to the solution orbits of (MD), we will further consider the *flow* $\Theta\colon \mathbb{R}_+ \times \mathcal{Z} \to \mathcal{Z}$ of (MD), which is simply the orbit of (MD) at time $t \in \mathbb{R}_+$ with an initial condition $z(0) = z \in \mathcal{Z}$. We then have the following notion of "asymptotic closeness" due to [BH96, BH95]:

Definition 4.1. $Z(t)$ is an *asymptotic pseudotrajectory* (APT) of (MD) if, for all $T > 0$, we have:

$$\lim_{t\to\infty} \sup_{0\leq h\leq T} \|Z(t+h) - \Theta_h(Z(t))\| = 0. \tag{4.7}$$

This comparison criterion is due to [BH96] and it plays a central role in our analysis. In words, it simply posits that $Z(t)$ eventually tracks the flow of (MD) with arbitrary accuracy over windows of arbitrary length; as a result, if Z_n is an APT of (MD), it is reasonable to expect its behavior to be closely correlated to that of (MD).

Our first result below makes this link precise. To state it, we will make the following assumptions:

$$\lim_{n\to\infty} B_n = 0, \tag{A5.1}$$

$$\textstyle\sum_{n=1}^{\infty} \gamma_n^2 \sigma_n^2 < \infty, \tag{A5.2}$$

both assumed to hold with probability 1. Under these blanket requirements, we have:

Theorem 4.1. *Suppose that* (RM) *is run with a step-size policy* γ_n *such that* $\sum_n \gamma_n = \infty$, $\lim_n \gamma_n = 0$, *and Assumptions* (A5.1)–(A5.2) *hold. Then, with probability* 1, *one of the following holds: a*) Z_n *is an APT of* (MD); *or b*) Z_n *is unbounded* (*and hence, non-convergent*).

A key challenge in proving Theorem 4.1 is that Assumptions (A5.1) and (A5.2) allow for very general error processes W_n in (RM), including cases where W_n is non-zero-mean ($b_n \neq 0$) and/or unbounded, either with positive probability or in all its moments (e.g., $\sup_n \mathbb{E}[\|W_n\|^q] = \infty$ for all $q \geq 2$). Because of this, earlier foundational results on asymptotic pseudotrajectories [BH96, Ben99] do not apply, and we need to employ a series of direct (sub)martingale convergence arguments to control the quadratic variation of Z_n. The precise argument relies on a pathwise version of the Burkholder–Davis–Gundy (BDG) maximal inequality [HH80], but the details are fairly involved so we defer them to the appendix.

4.4.3 Applications and examples

Applying Theorem 4.1 requires verifying Assumptions (A5.1) and (A5.2) for the algorithmic framework of Section 4.3. However, even though the noise $\mathsf{U}(z;\omega)$ in (SFO) is assumed zero-mean and sub-Gaussian, this *does not imply* that the generalized error term $W_n = U_n + b_n$ in Algorithms 4.1–4.5 enjoys the same guarantees. For example, the RM representation of Algorithms 4.2–4.4 has non-zero bias, while Algorithm 4.5 exhibits both non-zero bias *and* unbounded variance (the latter behaving as $\mathcal{O}(1/\delta_n^2)$ with $\delta_n \to 0$ as $n \to \infty$).

In the following proposition we prove that, for a wide range of parameters, Algorithms 4.1–4.5 indeed generate asymptotic pseudotrajectories of (MD).

Proposition 4.1. *Let* Z_n *be a sequence generated by any of the Algorithms 4.1–4.5. Assume further that:*

a) *For first-order methods* (*Algorithms 4.1–4.4*), *the algorithm is run with SFO feedback satisfying* (4.2) *and a step-size* γ_n *such that* $A/n \leq \gamma_n \leq B/(\log n)^{1+\varepsilon}$ *for some* $A, B, \varepsilon > 0$.

b) *For zeroth-order methods* (*Algorithm 4.5*), *the algorithm is run with parameters* γ_n *and* δ_n *such that* $\lim_n(\gamma_n + \delta_n) = 0$, $\sum_n \gamma_n = \infty$, *and* $\sum_n \gamma_n^2/\delta_n^2 < \infty$ (*e.g.*, $\gamma_n = 1/n$, $\delta_n = 1/n^{1/3}$).

Then, with probability 1*, one of the following holds: a*) Z_n *is an APT of* (MD)*; or b*) Z_n *is unbounded.*

Remark 4.4.1. We note that the requirements for (SFO) are closely linked to the assumptions for γ_n: for instance, one can remove the sub-Gaussian tail and impose only that $\mathsf{U}(z;\omega)$ in (SFO) is bounded in L^q for some $q \geq 2$, and the conclusion of Proposition 4.1 still holds as long as $\sum_n \gamma_n^{1+q/2} < \infty$.

We conclude this discussion with a remark on the boundedness clause for Z_n in Theorem 4.1 and Proposition 4.1. Clearly, if Z_n is unbounded, it cannot converge to a solution of (SP), so we need not go further in examining the failure of (RM) as a solution method. Still, for completeness, we provide in the appendix a coercivity condition for F which guarantees that Z_n is bounded with probability 1.

4.4.4 Convergence analysis

To proceed, it is important to recall that critical points alone cannot capture the broad spectrum of algorithmic behaviors when (MD) is not a gradient system: already in Fig. 4.1 we see a critical point surrounded by an ensemble of periodic orbits. To account for this considerably richer landscape, we will need some more notions from the theory of dynamical systems:

Definition 4.2. Let $\mathcal{S}$ be a nonempty compact subset of $\mathcal{Z}$. We then say that:

a) $\mathcal{S}$ is *invariant* if $\Theta_t(\mathcal{S}) \subseteq \mathcal{S}$ for all $t \geq 0$.

b) $\mathcal{S}$ is *attracting* if it is invariant and there exists a compact neighborhood $\mathcal{K}$ of $\mathcal{S}$ such that $\lim_{t\to\infty} \operatorname{dist}(\Theta_t(z), \mathcal{S}) = 0$ for all $z \in \mathcal{K}$.

c) $\mathcal{S}$ is *internally chain-transitive* (ICT) if it is invariant and $\Theta|_{\mathcal{S}}$ admits no attractors other than $\mathcal{S}$.

Heuristically, ICT sets are characterized by the property that any two points in such a set may be joined by a piecewise continuous chain of arbitrarily long segments of orbits of (MD) broken by arbitrarily small jump discontinuities. As such, they account for a wide range of invariant sets of (MD), ranging from stationary points and periodic orbits (cf. Fig. 4.1), to homoclinic loops (trajectories that join a unstable critical point to itself), limit cycles (isolated periodic orbits), and many others.

Our next result shows that, *with probability* 1*, any limit point of* (RM) *lies in an ICT set of F*:

Theorem 4.2. *Suppose that* (RM) *is run with a step-size sequence* γ_n *such that* $\sum_n \gamma_n = \infty$, $\lim_n \gamma_n = 0$. *If Assumptions* (A5.1) and (A5.2) *hold, then, with probability* 1*, we have: a*) Z_n *converges to an ICT set of F; or b*) Z_n *is unbounded* (*and hence, non-convergent*).

Corollary 4.1. *Let* Z_n *be a sequence generated by any of the Algorithms 4.1–4.5 with parameters as in Proposition 4.1. If* Z_n *is bounded, then, with probability* 1*, it converges to an ICT set of F.*

The proof of Theorem 4.2 builds on a series of deep results in [BH96]; see the appendix. In plain terms, the theorem asserts that any trajectory of (RM) is either unbounded or eventually converges to an ICT set, which is "infinitely close" to the long-term orbits of the mean dynamics (MD). In particular, it rules out *any other type of asymptotic behavior* (convergent or non-convergent).

In gradient systems – i.e., when $V = -\nabla f$ for some $f\colon \mathcal{Z} \to \mathbb{R}$ – the only ICT sets of (MD) are connected sets of critical points of f (for a detailed statement and proof, see the appendix). As a result, we can effortlessly conclude that any RM scheme exhibits the same asymptotic behavior in minimization problems: they converge to connected components of critical points of f.

At the other end of the spectrum, in the bilinear objective $F(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}\boldsymbol{y}$, we show in the appendix that *any* tuple $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^2$ belongs to an ICT set of F. The most crucial implication of this observation is that although there exist many non-critical convergent sets in bilinear games, *none of these can be an attractor*: for any bounded region $\mathcal{S}$, there always exists $z \notin \mathcal{S}$ such that, no matter how close z is to $\mathcal{S}$, the mean dynamics (MD) initialized at z will stay at a positive distance from $\mathcal{S}$.

Importantly, in the full space of min-max problems, the two settings described above are both outliers: mixing a gradient system with a bilinear component can give rise to *isolated periodic attractors* (limit cycles) and other forms of attracting ICT sets that cannot be observed in either gradient systems *or* bilinear games. Indeed, our final result in this section shows that, while (SEG) and/or (OG/PEG) might be capable of eliminating periodic orbits in bilinear games [DISZ18, GBV$^+$19, AMLJG19, LS19, MLZ$^+$19], these methods fail to escape *spurious (i.e., non-critical) attractors* arising in generic non-convex/non-concave objectives (see also Example 4.5.1 for a visual illustration). The formal statement is as follows:

Theorem 4.3. *Let $\mathcal{S}$ be an attractor of* (MD) *and fix some confidence level $\alpha > 0$. If γ_n is small enough and Assumptions* (A5.1) and (A5.2) *hold, there exists a neighborhood $\mathcal{U}$ of $\mathcal{S}$, independent of α, such that $\mathbb{P}(Z_n$ converges to $\mathcal{S} \mid Z_1 \in \mathcal{U}) \geq 1 - \alpha$.*

Corollary 4.2. *Let Z_n be a sequence generated by any of the Algorithms 4.1–4.5 with sufficiently small γ_n satisfying the conditions of Proposition 4.1. Then $\mathbb{P}(Z_n$ converges to $\mathcal{S} \mid Z_1 \in \mathcal{U}) \geq 1 - \alpha$.*

As we show in the next section, Corollary 4.2 can have catastrophic implications for the convergence of min-max optimization algorithms.

4.5 Spurious attractors: illustrations and examples

We now provide concrete examples of attracting ICT sets consisting *entirely* of non-critical points. For illustration purposes, we focus on the simple case $\mathcal{X} = \mathcal{Y} = \mathbb{R}$ with polynomial objectives; of course, all examples below can be suitably generalized to higher dimensions.

Despite their rudimentary character, these examples already reveal many unexpected phenomena that are unknown in the context of non-convex minimization (or convex-concave saddle-point problems).

▼ **Example 4.5.1** (Almost bilinear $\not\approx$ bilinear, instability $\not\approx$ escape)**.** Consider an arbitrarily small perturbation of a bilinear game:

$$F(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}\boldsymbol{y} + \varepsilon\phi(\boldsymbol{y}), \tag{4.8}$$

where $\varepsilon > 0$ and $\phi(\boldsymbol{y}) = \frac{1}{2}\boldsymbol{y}^2 - \frac{1}{4}\boldsymbol{y}^4$. This problem admits an unstable critical point at the origin; further, using a general criterion provided in the appendix, one can prove, for ε small enough, the existence of an *attracting* ICT set $\mathcal{S}$ in a neighborhood of the circle $\{z : \|z\|^2 = 4/3\}$. Thus, any of the RM schemes of Section 4.3 gets trapped by $\mathcal{S}$; see Fig. 4.2(a) for an illustration for (SEG).

This example brings two issues of existing studies to light. First, it shows that "almost bilinear games" can still trap many methods for solving exact bilinear games. Second, in contrast to minimization problems, the region around an unstable critical point can in fact be fully stable. Because of this, care needs to be taken when interpreting algorithms that are characterized as "locally avoiding unstable critical points", since they might be incapable of escaping their neighborhoods. ▲

▼ **Example 4.5.2** ("Forsaken" min-max points)**.** Suppose we apply Algorithms 4.1–4.5 to the objective

$$F(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}(\boldsymbol{y} - 0.5) + \phi(\boldsymbol{x}) - \phi(\boldsymbol{y}) \tag{4.9}$$

where $\phi(z) = \frac{1}{4}z^2 - \frac{1}{2}z^4 + \frac{1}{6}z^6$. This problem has a desirable min-max solution at $(\boldsymbol{x}^\star, \boldsymbol{y}^\star) = (0, 0.5)$. However, we prove in the appendix that there exist *two* spurious limit cycles that do not contain *any* critical point of F. Worse, the limit cycle closer to $(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$ is *unstable* and repels any trajectory that comes close to the solution; see Fig. 4.2(b) for an illustration for (SEG). Solutions that are "shielded" by spurious limit cycles in this way are unlikely to be visited by existing algorithms; to the best of our knowledge, no research has been conducted to tackle such problematic cases. ▲

▼ **Example 4.5.3** (Second-order methods)**.** Thanks to the efficient implementation of Hessian-gradient multiplications [Pea94], a popular second-order method for min-max optimization in machine learning is the *Hamiltonian descent* method [ALW19]. The idea is simply to run SGD on $f = \|\nabla F\|^2/2$, giving

$$Z_{n+1} = Z_n - \gamma_n J(Z_n)\nabla F(Z_n). \tag{HD}$$

As a (discretized) gradient system, our theory in Section 4.4 shows that (HD) does not possess ICT sets other than critical points. However, a serious issue of (HD) is that it ignores the *sign* of gradients, i.e., it does not distinguish between minimization and maximization. For this reason,

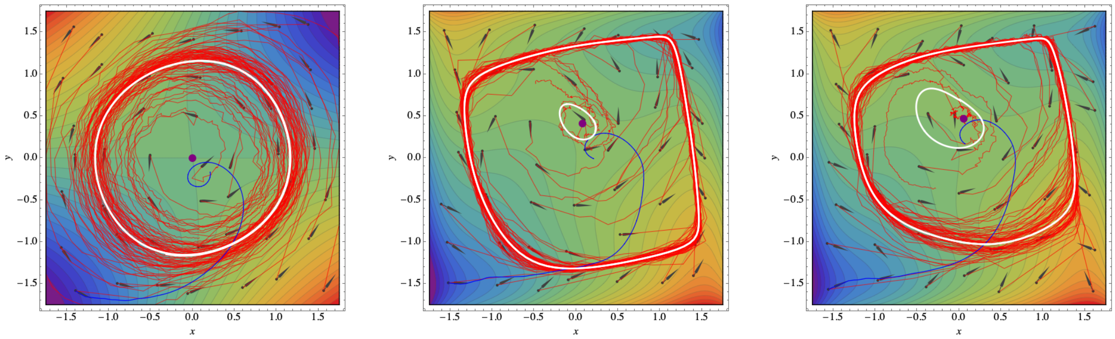


Figure 4.2: Spurious limits of min-max optimization algorithms. From left to right: (*a*) (SEG) for (4.8) with $\varepsilon = 0.01$; (*b*) "forsaken solutions" of (SEG); (*c*) "forsaken solutions" of SGA. The red curves present trajectories with different initialization; non-critical ICT sets are depicted in white; the blue curves represent an time-averaged sample orbit.

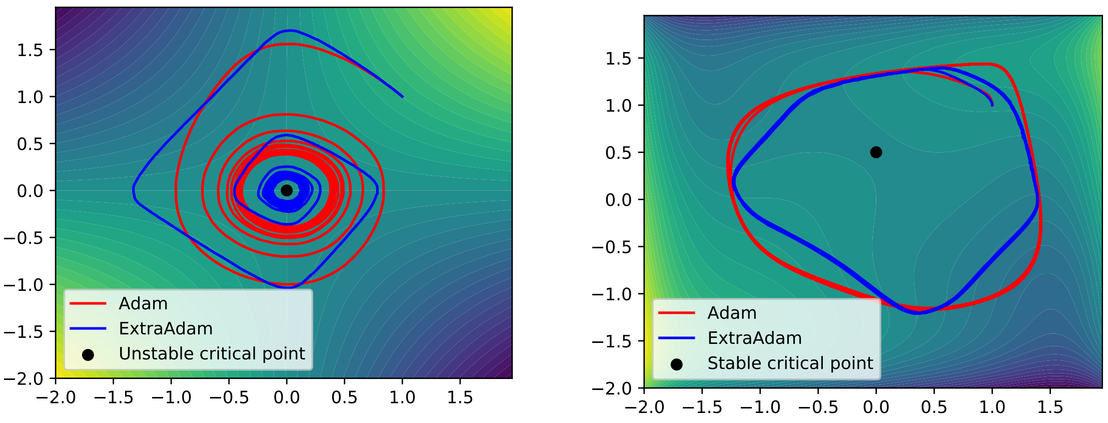
it has mostly been used as a *gradient penalty* scheme by mixing (HD) (or its variants) with (SGDA), giving rise to a number of other second-order methods such as *symplectic gradient adjustment* (SGA) [BRM$^+$18] and *consensus optimization* (ConO) [MNG17]. As in Section 4.3, one can cast these algorithms as RM schemes with $V(Z_n)$ replaced by $(I-\lambda J(Z_n))V(Z_n)$, where λ is the regularization parameter. The analysis can then proceed as in Section 4.4 by replacing (MD) with the appropriate continuous system.

Fig. 4.2(c) shows the spurious convergence of SGA with $\lambda = 0.2$ applied to (4.9). The ICT sets of SGA are only slightly different from Algorithms 4.1–4.5 and, in a certain precise sense, are perturbations thereof (so they suffer the same symptoms); see the appendix for more algorithms and details. ▲

We conclude with two remarks of a practical nature. First, Fig. 4.2 shows that the common tweak of *averaging* the iterates can force the trajectories to halt at non-critical points, and this convergence is by no means min-max optimal. To our knowledge, this provides the first explicit instances where training can get stuck even with non-vanishing gradients, a phenomenon often observed in training GANs.

Second, in Sections 4.5–4.5, we report the behaviors of popular *adaptive algorithms* in training GANs, including Adam [KB14] and its extra-gradient variant [GBV$^+$19], both with hyperparameters set to the default values in PyTorch. The result reveals a worrisome trend: while both Adam and ExtraAdam are able to somewhat mitigate the cycling of (4.8), this nonetheless comes at the price of converging to the *unstable* critical point $(0,0)$ (which is in fact a local max-min, the opposite of a desirable solution). On the other hand, as all RM schemes, both adaptive methods fail to reach the "forsaken" solutions in Example 4.5.2.

Finally, we stress that the purpose of examining these practical tweaks is *not* to prove that they will always fail (we have not performed extensive hyperparameter search). Rather, our aim is to point out that they cannot consistently serve as off-the-shelf solutions to the pathological ICT sets, and thus warrant a novel approach in studying min-max optimization problems.



(a) Adaptive algorithms for (4.8).

(b) Adaptive algorithms for (4.9).

5 Mixed Nash equilibria of min-max optimization problems

In Chapter 2, we learned that the *entropic mirrored descent* can be used to solve a *sampling* problem (i.e., finding the mixed Nash equilibrium) of finite games. In Chapter 3, we demonstrated how to draw samples from distributions that are defined over a continuum of variables.

The goal of this chapter is to show that, by combining the two techniques, we can solve min-max games with *any* number of strategies via a lifting trick. We further discuss the practical impact of our framework applied to important applications, such as *generative adversarial networks* (GANs) and *robust reinforcement learning* (RL).

5.1 Introduction

In Chapter 4, we have seen that many of the most challenging training problems in contemporary ML, including GANs and robust RL, amount to solving a min-max optimization problem (SP). In addition, we had rigorously shown that existing algorithms provably *fail* even in simple polynomial objectives. These negative results naturally prompt the question:

> *Does there exist a provably convergent algorithm for min-max games?*

The goal of this chapter is to answer the question in the affirmative, with the important tweak that we will modify the notion of "solution" from Chapter 4 (the so-called local pure *Nash equilibrium* (NE)) to *mixed NE*. We show that mixed NE can be solved via the classical *prox* methods in optimization, which include MD as a special case.

However, a downside of our solutions is the quadratic running time in the input parameters, rendering them impractical to training neural networks. To this end, we will further deduce a computationally efficient variant of our solutions, and showcase its empirical power on GANs and robust RL.

Notation. Throughout the chapter, we use $\boldsymbol{z}$ to denote a generic variable and $\mathcal{Z} \subseteq \mathbb{R}^d$ its domain. We denote the set of all (sufficiently regular) Borel probability measures on $\mathcal{Z}$ by $\mathcal{M}(\mathcal{Z})$, and the set of all (sufficiently regular)[1] functions on $\mathcal{Z}$ by $\mathcal{F}(\mathcal{Z})$. We write $\mathrm{d}\mu = \rho \mathrm{d}\boldsymbol{z}$ to mean that the density function of $\mu \in \mathcal{M}(\mathcal{Z})$ with respect to the Lebesgue measure is ρ. All integrals without specifying the measure are understood to be with respect to Lebesgue. For any objective of the form $\min_{\boldsymbol{x}} \max_{\boldsymbol{y}} F(\boldsymbol{x}, \boldsymbol{y})$, we say that $(\boldsymbol{x}_T, \boldsymbol{y}_T)$ is an $O\left(T^{-\frac{1}{2}}\right)$-NE if $\max_{\boldsymbol{x},\boldsymbol{y}} \{F(\boldsymbol{x}_T, \boldsymbol{y}) - F(\boldsymbol{x}, \boldsymbol{y}_T)\} = O\left(T^{-\frac{1}{2}}\right)$. Similarly we can define $O\left(T^{-1}\right)$-NE. The symbol $\|\cdot\|_{\mathbb{L}^\infty}$ denotes the $\mathbb{L}^\infty$-norm of functions, and $\|\cdot\|_{\mathrm{TV}}$ denotes the total variation norm of probability measures.

5.2 Mixed Nash equilibria and infinite-dimensional bi-affine games

We review standard results in game theory in Section 5.2.1, whose proof can be found in [Bub13a, Bub13b, Bub13c]. Section 5.2.2 performs a lifting trick to transform min-max objectives into the mixed NE formulation, and then relates the training of the min-max problem to the two-player game in Section 5.2.1, thereby suggesting to generalize the prox methods to infinite dimension.

5.2.1 Preliminary: finite bi-affine games

As a slight variant of the finite games in Chapter 2, consider the following two-player game with *finitely* many strategies:

$$\min_{\boldsymbol{p} \in \Delta_m} \max_{\boldsymbol{q} \in \Delta_n} \langle \boldsymbol{q}, \boldsymbol{a} \rangle - \langle \boldsymbol{q}, A\boldsymbol{p} \rangle, \tag{5.1}$$

where A is a payoff matrix, $\boldsymbol{a}$ is a vector, and $\Delta_d := \left\{\boldsymbol{z} \in \mathbb{R}^d \mid \sum_{i=1}^d z_i = 1\right\}$ is the probability simplex, representing the *mixed strategies* (i.e., probability distributions) over d pure strategies. A pair $(\boldsymbol{p}_{\mathrm{NE}}, \boldsymbol{q}_{\mathrm{NE}})$ achieving the min-max value in (5.1) is called a mixed NE.

Assume that the matrix A is too expensive to evaluate whereas the (stochastic) gradients of (5.1) are easy to obtain. Under such settings, a celebrated algorithm, the so-called *entropic Mirror Descent* (entropic MD), learns an $O\left(T^{-\frac{1}{2}}\right)$-NE: Let $h(\boldsymbol{z}) := \sum_{i=1}^d z_i \log z_i$ be the entropy function and $h^\star(\boldsymbol{y}) := \log \sum_{i=1}^d e^{y_i} = \sup_{\boldsymbol{z} \in \Delta_d} \left\{\langle \boldsymbol{z}, \boldsymbol{y} \rangle - h(\boldsymbol{z})\right\}$ be its Fenchel dual. For a learning rate η and an arbitrary vector $\boldsymbol{b} \in \mathbb{R}^d$, define the MD iterates as

$$\begin{aligned} \boldsymbol{z}' = \mathrm{MD}_\eta\left(\boldsymbol{z}, \boldsymbol{b}\right) \equiv \boldsymbol{z}' &= \nabla h^\star\left(\nabla h(\boldsymbol{z}) - \eta \boldsymbol{b}\right) \\ \equiv z_i' &= \frac{z_i e^{-\eta b_i}}{\sum_{i=1}^d z_i e^{-\eta b_i}}, \quad \forall 1 \leq i \leq d. \end{aligned} \tag{5.2}$$

The update rule takes linear time in dimension, which is highly scalable.

---

[1] See (D.1) and (D.2) for precise definitions.

Denote by $\bar{\boldsymbol{p}}_T := \frac{1}{T}\sum_{t=1}^T \boldsymbol{p}_t$ and $\bar{\boldsymbol{q}}_T := \frac{1}{T}\sum_{t=1}^T \boldsymbol{q}_t$ the ergodic average of two sequences $\{\boldsymbol{p}_t\}_{t=1}^T$ and $\{\boldsymbol{q}_t\}_{t=1}^T$. Then, with a properly chosen step-size η, the iterates

$$\begin{cases} \boldsymbol{p}_{t+1} = \mathrm{MD}_\eta\left(\boldsymbol{p}_t, -A^\top \boldsymbol{q}_t\right) \\ \boldsymbol{q}_{t+1} = \mathrm{MD}_\eta\left(\boldsymbol{q}_t, -\boldsymbol{a} + A\boldsymbol{p}_t\right) \end{cases}$$

come with the guarantee that $(\bar{\boldsymbol{p}}_T, \bar{\boldsymbol{q}}_T)$ is an $O\left(T^{-\frac{1}{2}}\right)$-NE. Moreover, a slightly more complicated algorithm, called the *entropic Mirror-Prox* (entropic MP) [Nem04], achieves faster rate than the entropic MD:

$$\begin{cases} \boldsymbol{p}_t = \mathrm{MD}_\eta\left(\tilde{\boldsymbol{p}}_t, -A^\top \tilde{\boldsymbol{q}}_t\right) \\ \boldsymbol{q}_t = \mathrm{MD}_\eta\left(\tilde{\boldsymbol{q}}_t, -\boldsymbol{a} + A\tilde{\boldsymbol{p}}_t\right) \\ \tilde{\boldsymbol{p}}_{t+1} = \mathrm{MD}_\eta\left(\tilde{\boldsymbol{p}}_t, -A^\top \boldsymbol{q}_t\right) \\ \tilde{\boldsymbol{q}}_{t+1} = \mathrm{MD}_\eta\left(\tilde{\boldsymbol{q}}_t, -\boldsymbol{a} + A\boldsymbol{p}_t\right) \end{cases}$$

implies that $(\bar{\boldsymbol{p}}_T, \bar{\boldsymbol{q}}_T)$ is an $O\left(T^{-1}\right)$-NE. If, instead of deterministic gradients, one uses unbiased stochastic gradients for entropic MD and MP, then both algorithms achieve $O\left(T^{-\frac{1}{2}}\right)$-NE in expectation.

5.2.2 Mixed strategy formulation for min-max games

For illustration, let us take Wasserstein GAN [ACB17] as an example, whereas the derivation in this section applies to any min-max objective. We perform a common bilinearization trick that dates back at least to the early literature in game theory [Gli52].

The training objective of Wasserstein GAN is

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{w}\in\mathcal{W}} \mathbb{E}_{X\sim\mathbb{P}_{\mathrm{real}}}[f_{\boldsymbol{w}}(X)] - \mathbb{E}_{X\sim\mathbb{P}_{\boldsymbol{\theta}}}[f_{\boldsymbol{w}}(X)], \tag{5.3}$$

where Θ is the set of parameters for the generator and $\mathcal{W}$ the set of parameters for the discriminator f, typically both taken to be neural nets.

The high-level idea of our approach is, instead of solving (5.3) directly, we focus on the *mixed strategy* formulation of (5.3). In other words, we consider the set of all probability distributions over Θ and $\mathcal{W}$, and we search for the optimal distribution that solves the following program:

$$\min_{\nu\in\mathcal{M}(\Theta)} \max_{\mu\in\mathcal{M}(\mathcal{W})} \mathbb{E}_{\boldsymbol{w}\sim\mu}\mathbb{E}_{X\sim\mathbb{P}_{\mathrm{real}}}[f_{\boldsymbol{w}}(X)] - \mathbb{E}_{\boldsymbol{w}\sim\mu}\mathbb{E}_{\boldsymbol{\theta}\sim\nu}\mathbb{E}_{X\sim\mathbb{P}_{\boldsymbol{\theta}}}[f_{\boldsymbol{w}}(X)]. \tag{5.4}$$

Define the function $g: \mathcal{W} \to \mathbb{R}$ by $g(\boldsymbol{w}) := \mathbb{E}_{X\sim\mathbb{P}_{\mathrm{real}}}[f_{\boldsymbol{w}}(X)]$ and the operator $G: \mathcal{M}(\Theta) \to \mathcal{F}(\mathcal{W})$ as $(G\nu)(\boldsymbol{w}) := \mathbb{E}_{\boldsymbol{\theta}\sim\nu, X\sim\mathbb{P}_{\boldsymbol{\theta}}}[f_{\boldsymbol{w}}(X)]$. Denoting $\langle \mu, h\rangle := \mathbb{E}_\mu h$ for any probability measure μ and function h, we may rewrite (5.4) as

$$\min_{\nu\in\mathcal{M}(\Theta)} \max_{\mu\in\mathcal{M}(\mathcal{W})} \langle \mu, g\rangle - \langle \mu, G\nu\rangle. \tag{5.5}$$

Algorithm 7: INFINITE-DIMENSIONAL ENTROPIC MD

```
Require: Initial distributions $\mu_1, \nu_1$, learning rate $\eta$
for $t = 1, 2, \ldots, T-1$ do
    $\nu_{t+1} = \mathrm{MD}_\eta\left(\nu_t, -G^\dagger \mu_t\right)$
    $\mu_{t+1} = \mathrm{MD}_\eta\left(\mu_t, -g + G\nu_t\right)$
end for
return $\bar{\nu}_T = \frac{1}{T}\sum_{t=1}^T \nu_t$ and $\bar{\mu}_T = \frac{1}{T}\sum_{t=1}^T \mu_t$.
```

Furthermore, the derivative (the analogue of gradient in infinite dimension) of (5.5) with respect to μ is simply $g - G\nu$, and the derivative of (5.5) with respect to ν is $-G^\dagger\mu$, where $G^\dagger : \mathcal{M}(\mathcal{W}) \to \mathcal{F}(\Theta)$ is the adjoint operator of G defined via the relation

$$\forall \mu \in \mathcal{M}(\mathcal{W}), \nu \in \mathcal{M}(\Theta), \quad \langle \mu, G\nu \rangle = \left\langle \nu, G^\dagger \mu \right\rangle. \tag{5.6}$$

One can easily check that $(G^\dagger\mu)(\boldsymbol{\theta}) := \mathbb{E}_{X\sim\mathbb{P}_{\boldsymbol{\theta}}, \boldsymbol{w}\sim\mu}[f_{\boldsymbol{w}}(X)]$ achieves the equality in (5.6).

To summarize, the mixed strategy formulation of Wasserstein GAN is (5.5), whose derivatives can be expressed in terms of g and G.

Now, observe that (5.5) is exactly the infinite-dimensional analogue of (5.1): The distributions over finite strategies are replaced with probability measures over a continuous parameter set, the vector $\boldsymbol{a}$ is replaced with a function g, the matrix A is replaced with a linear operator[2] G, and the gradients are replaced with derivatives. Based on Section 5.2.1, it is then natural to ask:

> *Can the entropic Mirror Descent and Mirror-Prox be extended to infinite dimension to solve (5.5)? Are there scalable implementations of these algorithms, at least approximately?*

We provide an affirmative answer to the first question in Section 5.3. The so obtained algorithms, nonetheless, are infinite-dimensional and requires infinite computational power to implement. For practical interest, in Section 5.4 we propose a sampling framework to approximate the infinite-dimensional prox methods in Section 5.3.

5.3 Infinite-dimensional prox methods

This section builds a rigorous infinite-dimensional formalism in parallel to the finite-dimensional prox methods and proves their convergence rates. We remark that these results are folklore among optimization experts and hence we adopt an informal presentation here, deferring all the technical details to the appendix. However, to our knowledge, they are not published until our paper [HLC19].

---

[2] The linearity of G trivially follows from the linearity of expectation.

Algorithm 8: Infinite-Dimensional Entropic MP

```
Require: Initial distributions $\tilde{\mu}_1, \tilde{\nu}_1$, learning rate $\eta$
for $t = 1, 2, \ldots, T$ do
    $\nu_t = \mathrm{MD}_\eta\left(\tilde{\nu}_t, -G^\dagger \tilde{\mu}_t\right)$
    $\mu_t = \mathrm{MD}_\eta\left(\tilde{\mu}_t, -g + G\tilde{\nu}_t\right)$
    $\tilde{\nu}_{t+1} = \mathrm{MD}_\eta\left(\tilde{\nu}_t, -G^\dagger \mu_t\right)$
    $\tilde{\mu}_{t+1} = \mathrm{MD}_\eta\left(\tilde{\mu}_t, -g + G\nu_t\right)$
end for
return $\bar{\nu}_T = \frac{1}{T}\sum_{t=1}^T \nu_t$ and $\bar{\mu}_T = \frac{1}{T}\sum_{t=1}^T \mu_t$.
```

We first recall the notion of derivative in infinite-dimensional spaces. A (nonlinear) functional $\Phi : \mathcal{M}(\mathcal{Z}) \to \mathbb{R}$ is said to possess a derivative at $\mu \in \mathcal{M}(\mathcal{Z})$ if there exists a function $\mathrm{d}\Phi(\mu) \in \mathcal{F}(\mathcal{Z})$ such that, for all $\mu' \in \mathcal{M}(\mathcal{Z})$, we have

$$\Phi(\mu + \epsilon\mu') = \Phi(\mu) + \epsilon\left\langle \mu', \mathrm{d}\Phi(\mu)\right\rangle + o(\epsilon).$$

Similarly, a (nonlinear) functional $\Phi^\star : \mathcal{F}(\mathcal{Z}) \to \mathbb{R}$ is said to possess a derivative at $h \in \mathcal{F}(\mathcal{Z})$ if there exists a measure $\mathrm{d}\Phi^\star(h) \in \mathcal{M}(\mathcal{Z})$ such that, for all $h' \in \mathcal{F}(\mathcal{Z})$, we have

$$\Phi^\star(h + \epsilon h') = \Phi^\star(h) + \epsilon\left\langle \mathrm{d}\Phi^\star(h), h'\right\rangle + o(\epsilon).$$

The most important functionals in this paper are the (negative) Shannon entropy

$$\mu \in \mathcal{M}(\mathcal{Z}), \quad \Phi(\mu) := \int \mathrm{d}\mu \log \frac{\mathrm{d}\mu}{\mathrm{d}\boldsymbol{z}}$$

and its Fenchel dual

$$h \in \mathcal{F}(\mathcal{Z}), \quad \Phi^\star(h) := \log \int e^h \mathrm{d}\boldsymbol{z}.$$

The first result of our paper is to show that, in direct analogy to (5.2), the infinite-dimensional MD iterates can be expressed as:

Theorem 5.1 (Infinite-Dimensional Mirror Descent, informal)**.** *For a learning rate η and an arbitrary function h, we can equivalently define*

$$\begin{aligned} \mu_+ = \mathrm{MD}_\eta\left(\mu, h\right) \equiv \mu_+ &= \mathrm{d}\Phi^\star\left(\mathrm{d}\Phi(\mu) - \eta h\right) \\ &\equiv \mathrm{d}\mu_+ = \frac{e^{-\eta h}\mathrm{d}\mu}{\int e^{-\eta h}\mathrm{d}\mu}. \end{aligned} \tag{5.7}$$

Moreover, most of the essential ingredients in the analysis of finite-dimensional prox methods can be generalized to infinite dimension.

See Theorem D.2 for precise statements and a long list of "essential ingredients of prox methods" generalizable to infinite dimension.

We are now ready introduce two "conceptual" algorithms for solving the mixed NE of Wasserstein GANs: The infinite-dimensional entropic MD in Algorithm 7 and MP in Algorithm 8.

Theorem 5.2 (Convergence Rates, informal). *Let* $\Phi(\mu) = \int \mathrm{d}\mu \log \frac{\mathrm{d}\mu}{\mathrm{d}z}$, *and let* $D(\cdot,\cdot)$ *be the relative entropy. Then, with a properly chosen step-size* η, *we have*

- *Assume that we have access to the deterministic derivatives. Then Algorithm 7 achieves* $O\left(T^{-\frac{1}{2}}\right)$*-NE, and Algorithm 8 achieves* $O\left(T^{-1}\right)$*-NE.*

- *Assume that we have access to stochastic derivatives such that the bias and the variance are small. Then Algorithm 7 with stochastic derivatives achieves* $O\left(T^{-\frac{1}{2}}\right)$*-NE in expectation, and Algorithm 8 with stochastic derivatives achieves* $O\left(T^{-\frac{1}{2}}\right)$*-NE in expectation.*

The precise statements of Theorem 5.2 and their proofs can be found in Appendix D.2.

5.4 A sampling framework for approximate infinite-dimensional prox methods

Section 5.4.1 reduces Algorithms 7–8 to a sampling routine [WT11] that has widely been used in machine learning. Section 5.4.2 proposes to further simplify the algorithms by summarizing a batch of samples by their mean.

For simplicity, we will only derive the algorithm for entropic MD; the case for entropic MP is similar but requires more computation. To ease the notation, we assume $\eta = 1$ throughout this section as η does not play an important role in the derivation below.

5.4.1 Implementable entropic MD: from probability measure to samples

We demonstrate how Algorithm 7 with stochastic derivatives can be reduced to simple sampling tasks. The reduction consists of three steps.

Step 1: Reformulating entropic mirror descent iterates

The definition of the MD iterate (5.7) relates the updated probability measure μ_{t+1} to the current probability measure μ_t, but it tells us nothing about the density function of μ_{t+1}, from which we want to sample. Our first step is to express (5.7) in a more tractable form. By recursively applying (5.7) and using Theorem D.2.10, we have, for some constants $C_1, \dots, C_{T-1}$,

$$\begin{aligned}
\mathrm{d}\Phi(\mu_T) &= \mathrm{d}\Phi(\mu_{T-1}) - \left(-g + G\nu_{T-1}\right) + C_{T-1} \\
&= \mathrm{d}\Phi(\mu_{T-2}) - \left(-g + G\nu_{T-2}\right) \\
&\qquad - \left(-g + G\nu_{T-1}\right) + C_{T-1} + C_{t-2} \\
&= \cdots \\
&= \mathrm{d}\Phi(\mu_1) - \left(-(T-1)g + G\sum_{s=1}^{T-1} \nu_s\right) + \sum_{s=1}^{T-1} C_s.
\end{aligned}$$

For simplicity, assume that μ_1 is uniform so that $\mathrm{d}\Phi(\mu_1)$ is a constant function. Then, by (D.7) and that $\mathrm{d}\Phi^\star\left(\mathrm{d}\Phi(\mu_T)\right) = \mathrm{d}\mu_T$, we see that the density function of μ_T is simply $\mathrm{d}\mu_T = \frac{\exp\{(T-1)g - G\sum_{s=1}^{T-1}\nu_s\}\mathrm{d}\boldsymbol{w}}{\int \exp\{(T-1)g - G\sum_{s=1}^{T-1}\nu_s\}\mathrm{d}\boldsymbol{w}}$. Similarly, we have $\mathrm{d}\nu_T = \frac{\exp\{G^\dagger \sum_{s=1}^{T-1}\mu_s\}\mathrm{d}\boldsymbol{\theta}}{\int \exp\{G^\dagger \sum_{s=1}^{T-1}\mu_s\}\mathrm{d}\boldsymbol{\theta}}$.

Step 2: Empirical approximation for stochastic derivatives

The derivatives of (5.5) involve the function g and operator G. Recall that g requires taking expectation over the real data distribution, which we do not have access to. A common approach is to replace the true expectation with its empirical average:

$$g(\boldsymbol{w}) = \mathbb{E}_{X\sim\mathbb{P}_{\mathrm{real}}}[f_{\boldsymbol{w}}(X)] \simeq \frac{1}{n}\sum_{i=1}^{n} f_{\boldsymbol{w}}(X_i^{\mathrm{real}}) \triangleq \hat{g}(\boldsymbol{w})$$

where X_i's are real data and n is the batch size. Clearly, $\hat{g}$ is an unbiased estimator of g.

On the other hand, $G\nu_t$ and $G^\dagger\mu_t$ involve expectation over ν_t and μ_t, respectively, and also over the fake data distribution $\mathbb{P}_{\boldsymbol{\theta}}$. Therefore, if we are able to draw samples from μ_t and ν_t, then we can again approximate the expectation via the empirical average:

$$\begin{aligned}
&\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \dots, \boldsymbol{\theta}^{(n')} \sim \nu_t,\ \left\{X_i^{(j)}\right\}_{i=1}^{n} \sim \mathbb{P}_{\boldsymbol{\theta}^{(j)}}, \\
&\hat{G}\nu_t(\boldsymbol{w}) \simeq \frac{1}{nn'}\sum_{i=1}^{n}\sum_{j=1}^{n'} f_{\boldsymbol{w}}\left(X_i^{(j)}\right),
\end{aligned}$$

and similarly,

$$\begin{aligned}
&\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}, \dots, \boldsymbol{w}^{(n')} \sim \mu_t,\ \{X_i\}_{i=1}^{n} \sim \mathbb{P}_{\boldsymbol{\theta}}, \\
&\hat{G}^\dagger\mu_t(\boldsymbol{\theta}) \simeq \frac{1}{nn'}\sum_{i=1}^{n}\sum_{j=1}^{n'} f_{\boldsymbol{w}^{(j)}}(X_i).
\end{aligned}$$

Now, assuming that we have obtained unbiased stochastic derivatives $-\sum_{s=1}^{t}\hat{G}^\dagger\mu_s$ and $\sum_{s=1}^{t}\left(-\hat{g} + \hat{G}\nu_s\right)$, how do we actually draw samples from μ_{t+1} and ν_{t+1}? Provided we can answer this question, then we can start with two easy-to-sample distributions (μ_1, ν_1), and then we will be able to draw samples from (μ_2, ν_2). These samples in turn will allow us to draw samples from (μ_3, ν_3), and so on. Therefore, it only remains to answer the above question. This leads us to:

Step 3: Sampling by stochastic gradient Langevin dynamics

For any probability distribution with density function $e^{-h}\mathrm{d}\boldsymbol{z}$, the Stochastic Gradient Langevin Dynamics (SGLD) [WT11] iterates as

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k - \gamma\hat{\nabla} h(\boldsymbol{z}_k) + \sqrt{2\gamma}\epsilon\xi_k, \tag{5.8}$$

where γ is the step-size, $\hat{\nabla} h$ is an unbiased estimator of ∇h, ϵ is the thermal noise, and $\xi_k \sim \mathcal{N}(0, I)$ is a standard normal vector, independently drawn across different iterations.

Suppose we start at (μ_1, ν_1). Plugging $h \leftarrow -\hat{G}^\dagger\mu_1$ and $h \leftarrow -\hat{g} + \hat{G}\nu_1$ into (5.8), we obtain, for $\{X_i\}_{i=1}^n \sim \mathbb{P}_{\boldsymbol{\theta}_k}$, $\{\boldsymbol{w}^{(j)}\}_{j=1}^{n'} \sim \mu_1$, standard normal ξ_k, ξ'_k, and $X_i^{\text{real}} \sim \mathbb{P}_{\text{real}}$, $\{\boldsymbol{\theta}^{(j)}\}_{j=1}^{n'} \sim \nu_1$, $\{X_i^{(j)}\} \sim \mathbb{P}_{\boldsymbol{\theta}^{(j)}}$, the following update rules:

$$\begin{aligned}
\boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k + \gamma\nabla_{\boldsymbol{\theta}}\left(\frac{1}{nn'}\sum_{i=1}^{n}\sum_{j=1}^{n'} f_{\boldsymbol{w}^{(j)}}(X_i)\right) + \sqrt{2\gamma}\epsilon\xi_k,\\
\boldsymbol{w}_{k+1} &= \boldsymbol{w}_k + \gamma\nabla_{\boldsymbol{w}}\left(\frac{1}{n}\sum_{i=1}^{n} f_{\boldsymbol{w}_k}(X_i^{\text{real}}) - \frac{1}{nn'}\sum_{i=1}^{n}\sum_{j=1}^{n'} f_{\boldsymbol{w}_k}\big(X_i^{(j)}\big)\right) + \sqrt{2\gamma}\epsilon\xi'_k.
\end{aligned}$$

The theory of [WT11, TTV16] states that, for large enough k, the iterates of SGLD above (approximately) generate samples according to the probability measures (μ_2, ν_2). We can then apply this process recursively to obtain samples from $(\mu_3, \nu_3), (\mu_4, \nu_4), \ldots (\mu_T, \nu_T)$. Finally, since the entropic MD and MP output the averaged measure $(\bar{\mu}_T, \bar{\nu}_T)$, it suffices to pick a random index $\hat{t} \in \{1, 2, \ldots, T\}$ and then output samples from $(\mu_{\hat{t}}, \nu_{\hat{t}})$.

Putting **Steps 1-3** together, we obtain Algorithms 10–11 in Appendix D.3.

Remark 5.4.1. In principle, any first-order sampling method is valid above. In the experimental section, we also use a RMSProp-preconditioned version of the SGLD [LCCC16].

5.4.2 Summarizing samples by averaging: a simple yet effective heuristic

Although Algorithms 10–11 are implementable, they are quite complicated and resource-intensive, as the total computational complexity is $O(T^2)$. This high complexity comes from the fact that, when computing the stochastic derivatives, we need to store all the historical samples and evaluate new gradients at these samples.

An intuitive approach to alleviate the above issue is to try to summarize each distribution by only *one* parameter. To this end, the mean of the distribution is the most natural candidate, which has also proven effective in practice. Moreover, the mean is often easier to acquire than the actual samples. For instance, computing the mean of distributions of the form $e^{-h}\mathrm{d}\boldsymbol{z}$, where h is a loss function defined by deep neural networks, has been empirically proven

Algorithm 9: MixedNE-LD

```
Input: step-size $\{\eta_t\}_{t=1}^T$, thermal noise $\{\epsilon_t\}_{t=1}^T$, warmup steps $\{K_t\}_{t=1}^T$, exponential damping factor $\beta$.
Initialize (randomly) $\omega_1, \theta_1$
for $t = 1, 2, \ldots, T-1$ do
    $\bar{\omega}_t, \omega_t^{(1)} \leftarrow \omega_t$ ; $\bar{\theta}_t, \theta_t^{(1)} \leftarrow \theta_t$
    for $k = 1, 2, \ldots, K_t$ do
        $\xi, \xi' \sim \mathcal{N}(0, I)$
        $\theta_t^{(k+1)} \leftarrow \theta_t^{(k)} + \eta_t \nabla_\theta \widehat{h\left(\theta_t^{(k)}, \omega_t\right)} + \sqrt{2\eta_t}\epsilon_t \xi$
        $\omega_t^{(k+1)} \leftarrow \omega_t^{(k)} - \eta_t \nabla_\omega \widehat{h\left(\theta_t, \omega_t^{(k)}\right)} + \sqrt{2\eta_t}\epsilon_t \xi'$
        $\bar{\omega}_t \leftarrow \left(1-\beta\right)\bar{\omega}_t + \beta\omega_t^{(k+1)}$
        $\bar{\theta}_t \leftarrow \left(1-\beta\right)\bar{\theta}_t + \beta\theta_t^{(k+1)}$
    end for
    $\omega_{t+1} \leftarrow \left(1-\beta\right)\omega_t + \beta\bar{\omega}_t$
    $\theta_{t+1} \leftarrow \left(1-\beta\right)\theta_t + \beta\bar{\theta}_t$
end for
return $\omega_T, \theta_T$.
```

successful in [CCS$^+$17, COO$^+$18, DR18] via SGLD. In this paper, we adopt the same approach as in [CCS$^+$17] where we use exponential damping (the β term in Algorithm 9) to increase stability. Algorithm 9, dubbed the *MixedNE-LD*, shows how to encompass this idea into entropic MD; the pseudocode for the similar *Mirror-Prox-GAN* can be found in Algorithm 12 of Appendix D.3.

5.5 Empirical justification of MixedNE-LD

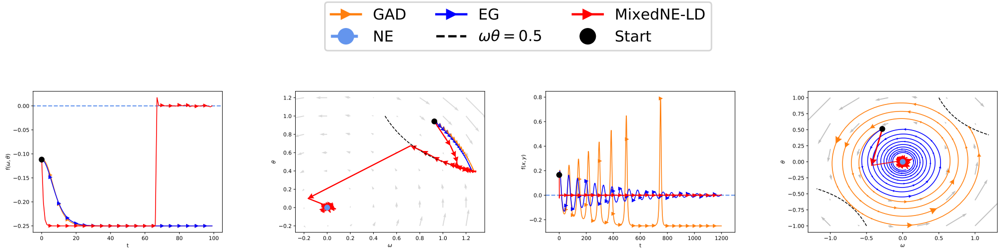
The goal of the present section is to demonstrate that solving (SP) as in its mixed NE formulation has superior performance over methods that seek pure NE for non-convex/non-concave objectives. We do so by providing theoretical and empirical justifications on several simple, yet nontrivial, low-dimensional examples. Since it is customary to maximize the reward function in RL, we will abuse the notation and consider the following formulation of robust RL:

$$\max_{\boldsymbol{x}\in\mathcal{X}} \min_{\boldsymbol{y}\in\mathcal{Y}} F(\boldsymbol{x}, \boldsymbol{y}).$$

Pseudocodes for all algorithms in the section and the omitted proofs can be found in Appendix D.4.

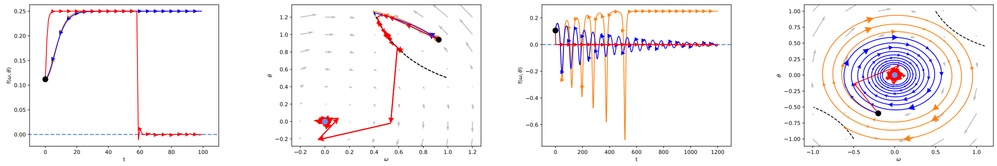
5.5.1 Existing algorithms

We will consider three algorithmic frameworks:



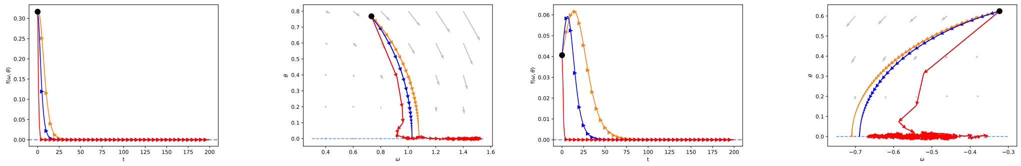
(a) $F(\boldsymbol{x}_t, \boldsymbol{y}_t)$, away from NE. (b) $(\boldsymbol{x}_t, \boldsymbol{y}_t)$, away from NE. (c) $F(\boldsymbol{x}_t, \boldsymbol{y}_t)$, close to NE. (d) $(\boldsymbol{x}_t, \boldsymbol{y}_t)$, close to NE.

Figure 5.1: $F(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^2\boldsymbol{y}^2 - \boldsymbol{x}\boldsymbol{y}$. The NE is $(0,0)$ with reward value 0. The dashed curve $\boldsymbol{x}\boldsymbol{y} = 0.5$ describe all stationary points that are *not* NE. (a), (b) shows the objective value and the training dynamics when initializing far away from NE. (c), (d) shows the objective value and the training dynamics when $(\boldsymbol{x}_1, \boldsymbol{y}_1)$ is initializing close to NE.



(a) $F(\boldsymbol{x}_t, \boldsymbol{y}_t)$, away from NE. (b) $(\boldsymbol{x}_t, \boldsymbol{y}_t)$, away from NE. (c) $F(\boldsymbol{x}_t, \boldsymbol{y}_t)$, close to NE. (d) $(\boldsymbol{x}_t, \boldsymbol{y}_t)$, close to NE.

Figure 5.2: $F(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}y - \boldsymbol{x}^2\boldsymbol{y}^2$. The NE is $(0,0)$ with reward value 0. The dashed curve $\boldsymbol{x}\boldsymbol{y} = 0.5$ are stationary points that are *not* NE. (a), (b) shows the objective value and the training dynamics when initializing far away from NE. (c), (d) shows the objective value and the training dynamics when initializing close to NE.



(a) $F(\boldsymbol{x}_t, \boldsymbol{y}_t)$, away from NE. (b) $(\boldsymbol{x}_t, \boldsymbol{y}_t)$, away from NE. (c) $F(\boldsymbol{x}_t, \boldsymbol{y}_t)$, close to NE. (d) $(\boldsymbol{x}_t, \boldsymbol{y}_t)$, close to NE.

Figure 5.3: $F(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^2\boldsymbol{y}^2$. The NE are represented with the line $\{(\boldsymbol{x}, 0) \mid \boldsymbol{x} \text{ arbitrary}\}$ with reward value 0. (a), (b) shows the objective value and the training dynamics when initializing far away from NE. (c), (d) shows the objective value and the training dynamics when initializing close to NE.

1. GAD: Finding pure NE via **G**radient **a**scent-**d**escent.
2. EG: Finding pure NE via **E**xtra-**g**radient methods.
3. MixedNE-LD: Finding mixed NE via Algorithm 9.

Here, GAD can be considered as the most naïve approach, while EG presents the state-of-the-art for GANs and robust RL.

It is common in practice to asymptotically decrease the step-size for GAD and EG to 0. According to the theory in Chapter 4, these first-order methods with vanishing step-size behave asymptotically the same as their continuous-time counterpart, i.e., (MD):

$$\begin{bmatrix} \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}(t) \\ \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}(t) \end{bmatrix} = \begin{bmatrix} \nabla_{\boldsymbol{x}} F(\boldsymbol{x},\boldsymbol{y}) \\ -\nabla_{\boldsymbol{y}} F(\boldsymbol{x},\boldsymbol{y}) \end{bmatrix} \tag{5.9}$$

Moreover, this result is robust to gradient noise, and so applies to stochastic variants of GAD and EG. Therefore, we will henceforth focus on (5.9) in our theory.

In Section 4.5, we have presented a number of problematic cases where state-of-the-art algorithms *provably* fail. In Appendix D.4.2, we show that our Algorithm 9 can somehow alleviate the failure of existing methods when complicated ICT sets arise. However, in the following section, we shall demonstrate another important feature of Algorithm 9 over existing methods: escaping undesirable stationary points in non-concave/non-convex objectives.

5.5.2 Degree-2 polynomials: stationary points vs. NE

Suppose that the objective F in (SP) is non-concave/non-convex in d directions. Since in practice one rarely acquires information higher than second-order, we will only consider quadratic local approximations of F. Finally, let us consider optimizing each dimension separately, each leading to a 2-dimensional subproblem.

We will show, in Theorem 5.3 below, that even under this extremely simplified setting, and under simple non-convexity as in (5.10) or (5.11), existing approaches can only succeed if the initialization is close enough to the equilibrium *along every direction.* As a result, the probability of successful training for existing algorithms will be exponential small in the number of non-convex non-concave directions.

We now construct nontrivial examples where there exist stationary points that are *not* NE. To this end, we may simply use the degree-2 polynomials:

$$\max_{\boldsymbol{x}\in[-2,2]} \min_{\boldsymbol{y}\in[-2,2]} F(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}^2\boldsymbol{y}^2 - \boldsymbol{x}\boldsymbol{y} \tag{5.10}$$

and

$$\max_{\boldsymbol{x}\in[-2,2]} \min_{\boldsymbol{y}\in[-2,2]} F(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}\boldsymbol{y} - \boldsymbol{x}^2\boldsymbol{y}^2. \tag{5.11}$$

The constraint interval $[-2,2]$ is included only for ease of presentation; it has no impact on our conclusion. Moreover, the following facts can be readily verified:

- The pure and mixed NE are the same: $(0,0)$.

- The curve $\{(\boldsymbol{x},\boldsymbol{y}) \mid \boldsymbol{x}\boldsymbol{y} = 0.5\}$ presents stationary points that are *not* NE.

5.5.3 Main result

We now present the main result in this section.

Theorem 5.3. *Consider the (continuous-time) GAD and EG dynamics* (5.9) *where* $F(\boldsymbol{x},\boldsymbol{y})$ *is either* (5.10) *or* (5.11). *Suppose that the initial point* $(\boldsymbol{x}(0),\boldsymbol{y}(0))$ *is far away from NE:* $\boldsymbol{x}(0)\cdot\boldsymbol{y}(0) > 0.5$. *Then* (5.9) *converges to a non-equilibrium stationary point on* $\{\boldsymbol{xy} = 0.5\}$.

On the other hand, even when initialized at a stationary point such that $\boldsymbol{x}_1\cdot\boldsymbol{y}_1 = 0.5$, *the MixedNE-LD still decreases the distance to NE in expectation:*

$$\mathbb{E}\boldsymbol{x}_3\cdot\boldsymbol{y}_3 = \boldsymbol{x}_1\boldsymbol{y}_1 - 4\eta^2\left(\eta\left(\boldsymbol{x}_1^2+\boldsymbol{y}_1^2\right)+14\eta^2\right) < \boldsymbol{x}_1\cdot\boldsymbol{y}_1 \tag{5.12}$$

where η *is the step-size, and the expectation is over the randomness of the algorithm.*

In words, depending on the initialization, the (continuous-time) training dynamics of GAD and EG will either get trapped by non-equilibrium stationary points, or converge to NE. In contrast, the MixedNE-LD is always able to escape non-equilibrium stationary points in expectation.

Figs. 5.1–5.2 demonstrate the empirical behavior of the three algorithms, which is in perfect accordance with the theory. When initialized far away from NE, Figs. 5.1(a)–5.1(b) and Figs. 5.2(a)–5.2(b) show that GAD and EG get trapped by local stationary points, while MixedNE-LD is able to escape after staying a few iterations near the non-equilibrium states. On the other hand, if initialized sufficiently close to NE, then EG tends to perform better than GAD, as indicated by previous work; see Figs. 5.1(c)–5.1(d) and Figs. 5.2(c)–5.2(d).

Finally, one can ask whether the negative results for GAD and EG are sensitive to the choice of step-size. For instance, we have implemented the vanilla GAD and EG, while in practice one always uses adaptive step-size based on approximate second-order information [DHS11, KB14]. However, our next theorem shows that, even with *perfect* second-order information, the training dynamics of GAD and EG still are unable to escape stationary points.

Theorem 5.4. *Consider the Newton's dynamics for solving either* (5.10) *or* (5.11)*:*

$$\begin{bmatrix} \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}(t) \\ \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}(t) \end{bmatrix} = \begin{bmatrix} \nabla^2_{\boldsymbol{x}}F(\boldsymbol{x},\boldsymbol{y}) & 0 \\ 0 & \nabla^2_{\boldsymbol{y}}F(\boldsymbol{x},\boldsymbol{y})] \end{bmatrix}^{-1} \begin{bmatrix} \nabla_{\boldsymbol{x}}F(\boldsymbol{x},\boldsymbol{y}) \\ -\nabla_{\boldsymbol{y}}F(\boldsymbol{x},\boldsymbol{y}) \end{bmatrix}. \tag{5.13}$$

Then we have $\boldsymbol{x}(t)\cdot\boldsymbol{y}(t) = \boldsymbol{x}(0)\cdot\boldsymbol{y}(0)$.

A consequence of Theorem 5.4 is that if we initialize at any point such that $\boldsymbol{x}(0)\cdot\boldsymbol{y}(0) \neq 0$, the training dynamics will remain far away from $(0,0)$, which is the desired NE. Indeed, in Section 5.7, we shall see that MixedNE-LD outperforms GAD and EG even with adaptivity.

5.5.4 A digression: sampling vs. optimization

We would like to demonstrate an additional intriguing behavior of the sampling nature of MixedNE-LD, which we deem as a benefit over deterministic optimization algorithms. Consider the following min-max problem:

$$\max_{\boldsymbol{x}\in[-2,2]} \min_{\boldsymbol{y}\in[-2,2]} F(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}^2\boldsymbol{y}^2. \tag{5.14}$$

This is a simple objective where the stationary points $\{(\boldsymbol{x},0) \mid \boldsymbol{x} \in [-2,2]\}$ are all NE. Consequently, both GAD and EG succeed in finding an NE, regardless of the initialization; see Fig. 5.3.

The MixedNE-LD, nonetheless, does something slightly more than finding an NE: The MixedNE-LD *explores* among all the NE, inducing a *distribution* on the set of all equilibria; see Figs. 5.3(b)–5.3(d). As exploration is a desirable property in RL, our experiments illustrate yet another advantage of pursuing the mixed NE over pure NE.

5.6 Experiment I: GANs

We now provide empirical evidence demonstrating that our sampling algorithms for mixed NE consistently outperform existing methods that seek pure NE. In this section, we focus on GANs.

We use visual quality of the generated images to evaluate different algorithms. We avoid reporting numerical metrics, as recent studies [BS18, Bor18, LKM+18] suggest that these metrics might be flawed. Setting of the hyperparameters and more auxiliary results can be found in Appendix D.5.

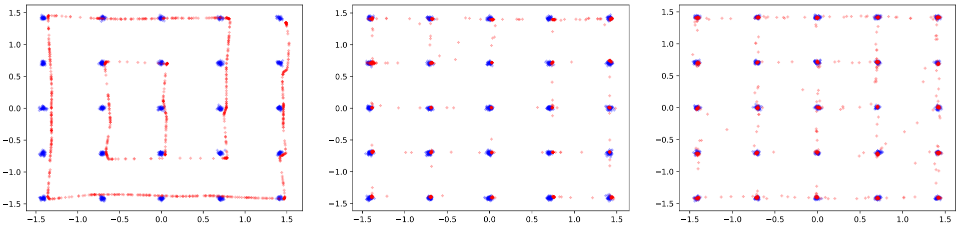
5.6.1 Synthetic data

We repeat the synthetic setup as in [GAA+17]. The tasks include learning the distribution of 8 Gaussian mixtures, 25 Gaussian mixtures, and the Swiss Roll. For both the generator and discriminator, we use two MLPs with three hidden layers of 512 neurons. We choose SGD and Adam as baselines, and we compare them to MixedNE-LD and Mirror-Prox-GAN. We also incorporate two contemporary algorithms, namely the Optimistic Adam [DISZ18] and (Simultaneous) Extra-Adam [GBV+19]. The step-sizes for all algorithms are determined via parameter sweeping.

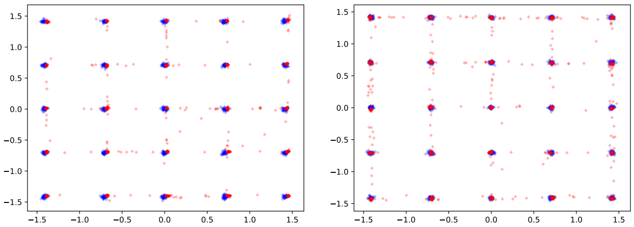
All algorithms are run up to 10^5 iterations[3]. The results of 25 Gaussian mixtures are shown in Fig. 5.4; An enlarged figure of 25 Gaussian Mixtures and other cases can be found in Appendix D.5.1.

---

[3] One iteration here means using one mini-batch of data. It does not correspond to the T in our algorithms, as there might be multiple SGLD iterations within each time step t.



(a) SGD (b) Optimistic Adam (c) MixedNE-LD



(d) Adam (e) Mirror-Prox-GAN

Figure 5.4: Fitting 25 Gaussian mixtures up to 10^5 iterations. Blue dots represent the true distribution and red ones are from the trained generator.

As Fig. 5.4 shows, SGD performs poorly in this task, while the other algorithms yield reasonable results. However, compared to Adam, MixedNE-LD and Mirror-Prox-GAN fit the true distribution better in two aspects. First, the modes found by MixedNE-LD and Mirror-Prox-GAN are more accurate than the ones by Adam, Optimistic Adam, and Extra-Adam, which are perceptibly biased. Second, MixedNE-LD and Mirror-Prox-GAN perform much better in capturing the variance (how spread the blue dots are), while Adam-based algorithms tend to collapse to modes. These observations are consistent throughout the synthetic experiments; see Appendix D.5.1.

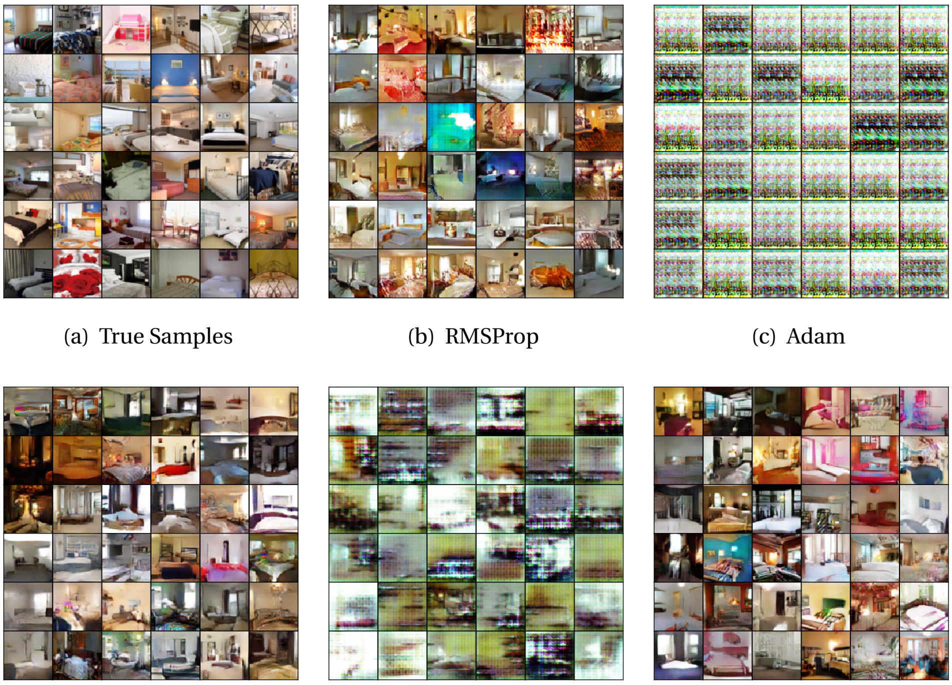
We also report that MixedNE-LD and Mirror-Prox-GAN are not only better in terms of solution quality, but also in speed: see Fig. D.4 in Appendix D.5.1.

5.6.2 Real data

For real images, we use the `LSUN bedroom` dataset [YSZ+15]. We have also conducted a similar study with `MNIST`; more results can be found in Appendix D.5.2.

We use the same architecture (DCGAN) as in [RMC15] with batch normalization. As the networks become deeper in this case, the gradient magnitudes differ significantly across different layers. As a result, non-adaptive methods such as SGD or SGLD do not perform well in this scenario. To alleviate such issues, we replace SGLD by the RMSProp-preconditioned SGLD [LCCC16] for our sampling routines. For baselines, we consider two adaptive gradient methods: RMSprop and Adam.

We also include the Extra-Adam, along with its alternated version [GBV+19]. However, we


(a) True Samples (b) RMSProp (c) Adam

(d) MixedNE-LD (e) Simultaneous Extra-Adam (f) Alternated Extra-Adam

Figure 5.5: Dataset `LSUN bedroom`, 10^5 iterations.

remark that the theory of [GBV+19] only provides motivations for simultaneous updates, and Alternated Extra-Adam should be considered as a heuristics. We drop Optimistic Adam in the this experiment since it is reported by [GBV+19] to be outperformed by Extra-Adam.

Fig. 5.5 shows the results at the 10^5th iteration, where step-sizes for all algorithms are determined by parameter sweeping. The RMSProp, Alternated Extra-Adam and MixedNE-LD produce images with reasonable quality, while Adam and simultaneous Extra-Adam fail to learn the distributions. The visual quality of Alternated Extra-Adam and MixedNE-LD are comparable, and are better than RMSProp, as RMSProp sometimes generates blurry images (the $(3,3)$- and $(1,5)$-th entry of Fig. D.6.(b)).

It is worth mentioning that Adam can learn the true distribution at intermediate iterations, but later on suffers from mode collapse and finally degenerates to noise; see Appendix D.5.2.

5.7 Experiment II: robust reinforcement learning

In this section, we demonstrate the effectiveness of using the MixedNE-LD framework to solve the robust RL problem.

5.7.1 Off-policy (DDPG) experiments

As a case study, we consider the *noisy robust Markov decision process* NR-MDP setting with $\delta = 0.1$ [TEM19]. This setting can cover only the changes in the transition dynamics that can be simulated via the changes in the action. In the H_∞ control literature [DFT13, MD05], an equivalence between environmental and action robustness has already been noted. The NR-MDP setting cannot handle:

1. the adversarial disturbances considered in [PDSG17], as the action spaces of both the agent and adversary are same in the NR-MDP setting.
2. the feature changes like style, and illumination.

Nevertheless, the MixedNE-LD framework applies to general two-player Markov Games as well.

Two-Player DDPG: We design a two-player variant of DDPG [LHP+15] algorithm by adapting the Algorithm 9. As opposed to standard DDPG, in two-player DDPG two actor networks output two deterministic policies, the protagonist and adversary policies, denoted by μ_θ and ν_ω. The critic is trained to estimate the Q-function of the joint-policy. The gradients of the protagonist and adversary parameters are given in Proposition 5 of [TEM19]. The resulting algorithm is given in Algorithm 14.

We compare the performance of our algorithm against the baseline algorithm proposed in [TEM19] (see Algorithm 15 with GAD). [TEM19] have suggested a training ratio of 1 : 1 for actors and critic updates. Note that the action noise is injected while collecting transitions for the replay buffer. In [FvHM18], authors noted that the action noise drawn from the Ornstein-Uhlenbeck [UO30] process offered no performance benefits. Thus we also consider uncorrelated Gaussian noise. In addition to the baseline from [TEM19], we have also considered another baseline, namely Algorithm 15 with Extra-Adam [GBV+19].

Setup: We evaluate the performance of Algorithm 14 and Algorithm 15 (with GAD and Extra-Adam) on standard continuous control benchmarks available on OpenAI Gym [BCP+16] utilizing the MuJoCo environment [TET12]. Specifically, we benchmark on eight tasks: Walker, Hopper, Half-Cheetah, Ant, Swimmer, Reacher, Humanoid, and InvertedPendulum. Details of these environments can be found in [BCP+16] and on the GitHub website.

The Algorithm 14 implementation is based on the codebase from [TEM19]. For all the algorithms, we use a two-layer feedforward neural network structure of (64, 64, tanh) for both actors (agent and adversary) and critic. The optimizer we use to update the critic is Adam [KB15] with a learning rate of 10^{-3}. The target networks are soft-updated with $\tau = 0.999$.

For the GAD baseline, the actors are trained with RMSProp optimizer. For our algorithm

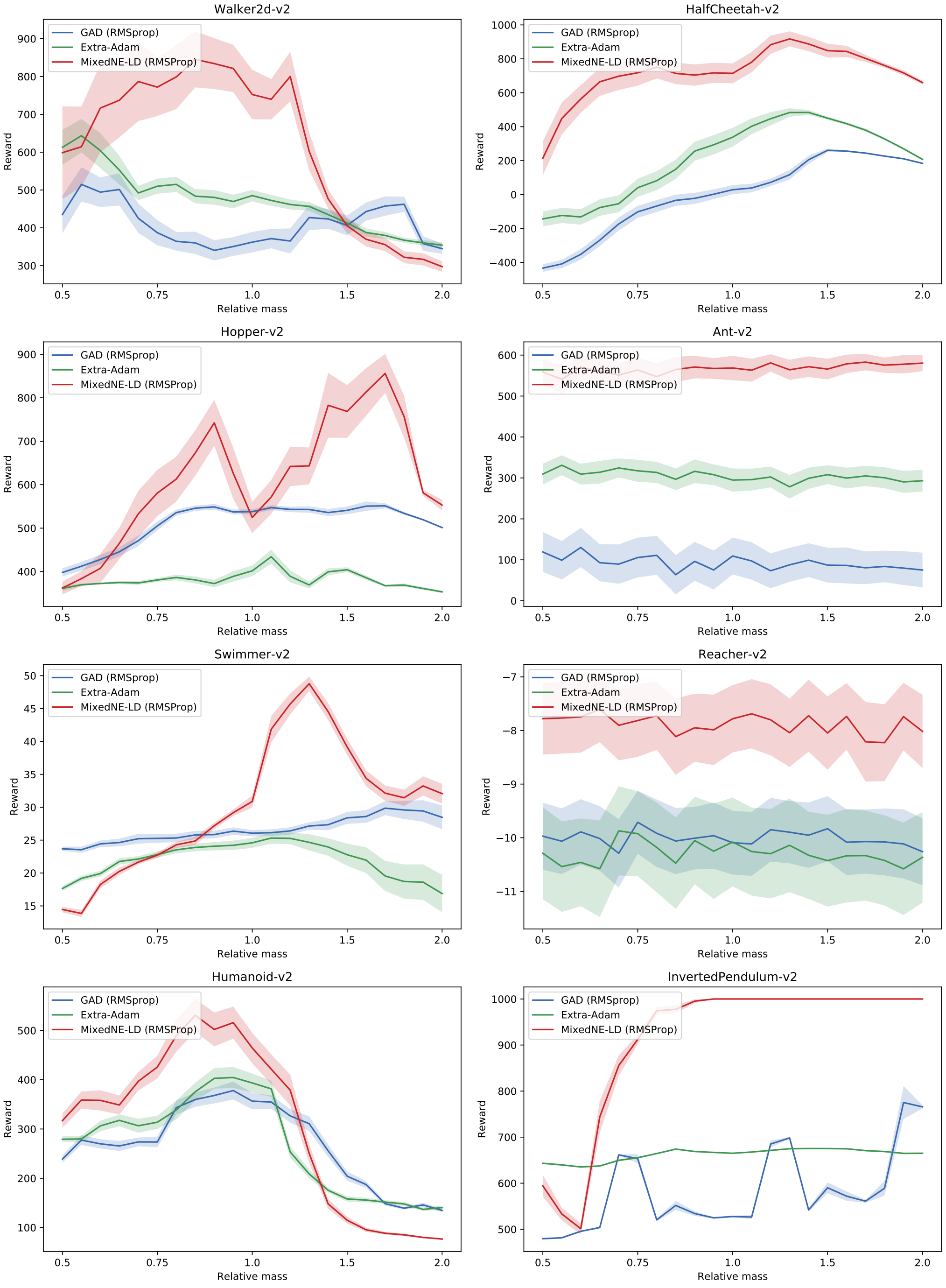


Figure 5.6: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed without adversarial perturbations, on a range of mass values not encountered during training.

(MixedNE-LD), the actors are updated according to Algorithm 9 with warmup steps $K_t = \min\{15, \lfloor (1+10^{-5})^t \rfloor\}$, and thermal noise $\sigma_t = \sigma_0 \times (1 - 5 \times 10^{-5})^t$. The hyperparameters that are not related to exploration (see Table D.3) are identical to all the algorithms that are

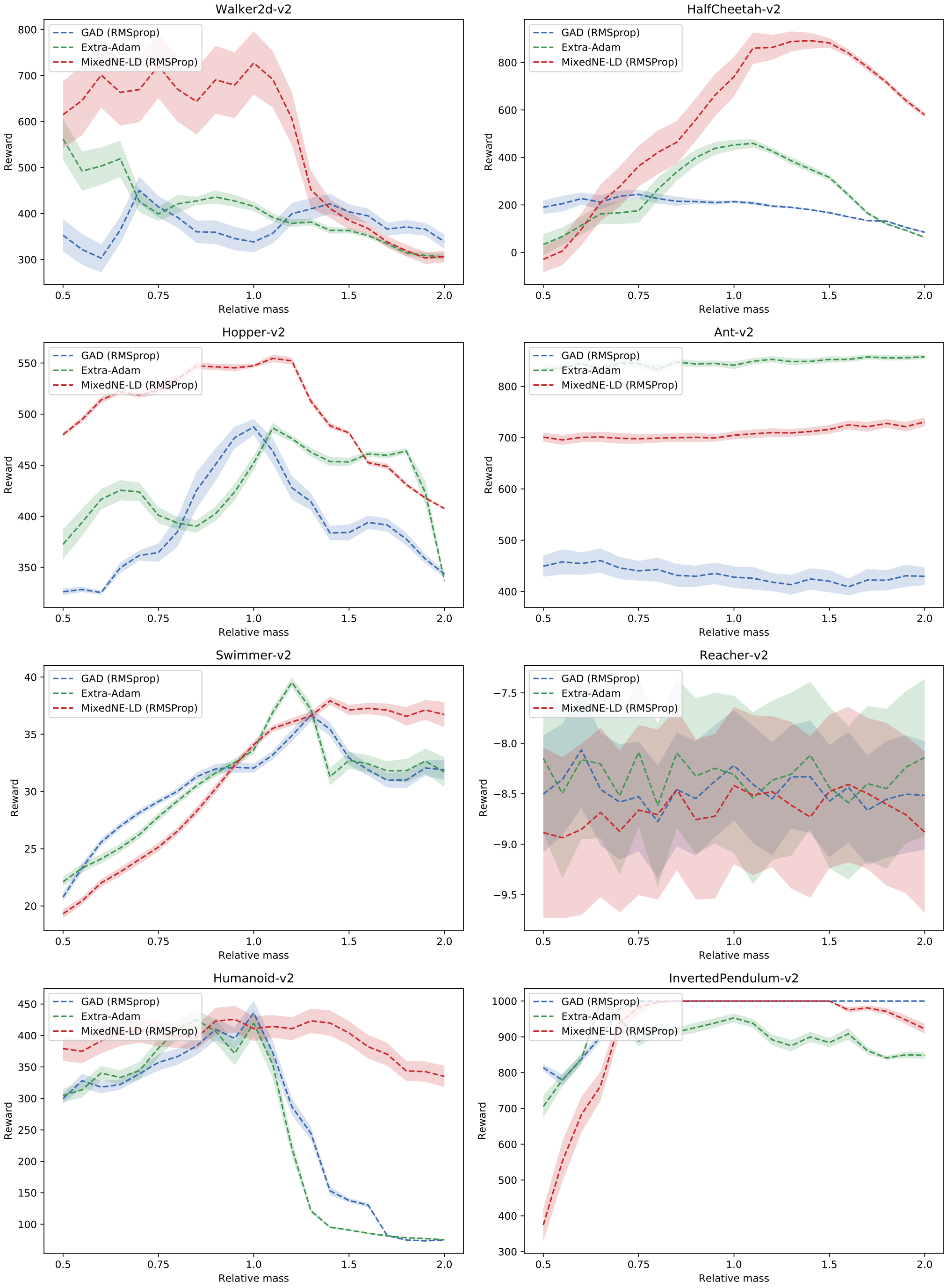


Figure 5.7: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed without adversarial perturbations, on a range of mass values not encountered during training.

compared.

And we tuned only the exploration-related hyperparameters (for all the algorithms) by grid

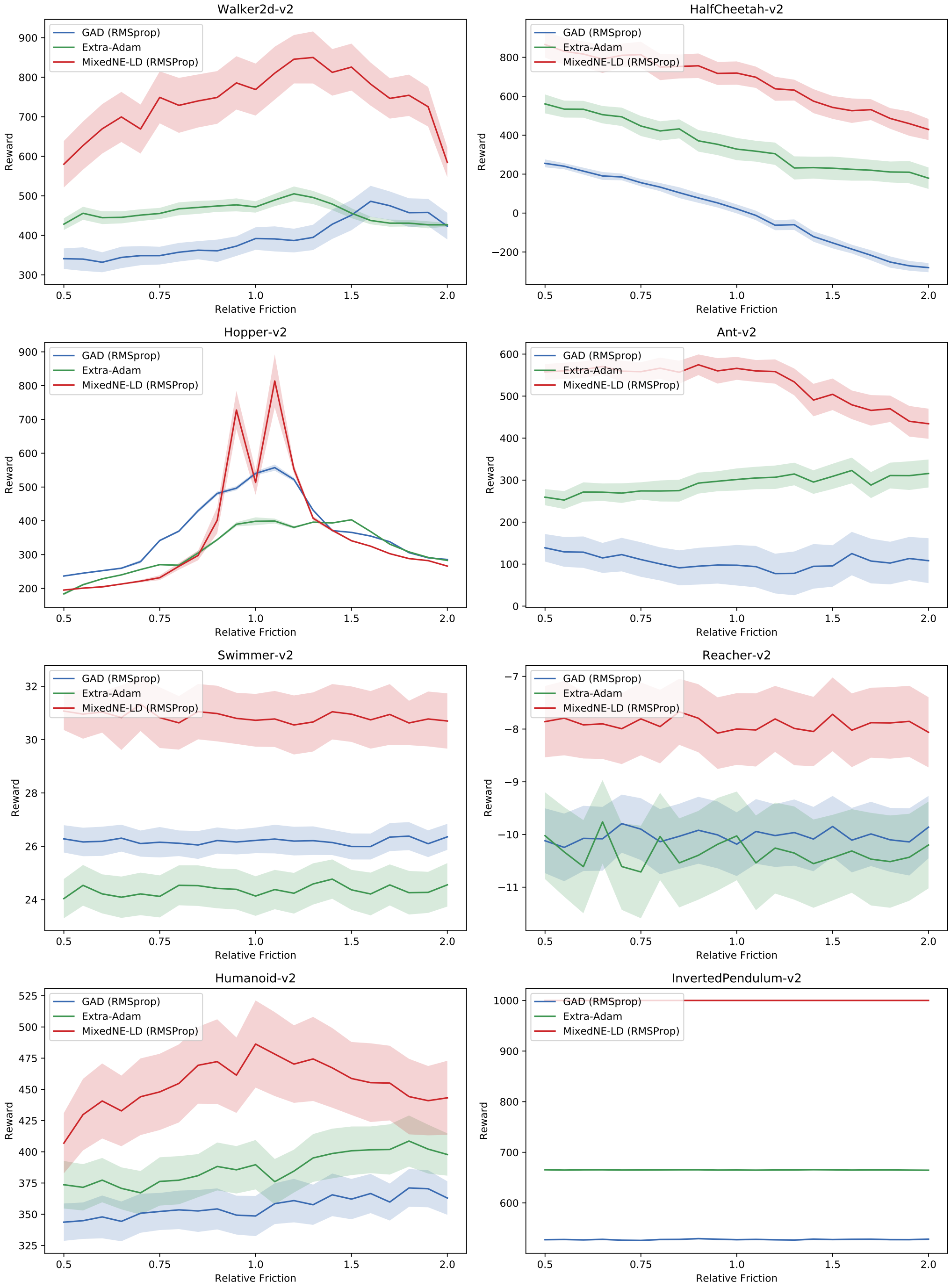


Figure 5.8: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed without adversarial perturbations, on a range of friction values not encountered during training.

search: (a) Algorithm 14 with $(\sigma_0, \sigma) \in \left\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\right\} \times \{0, 0.01, 0.1, 0.2, 0.3, 0.4\}$; (b) Algorithm 15 with $\sigma \in \{0, 0.01, 0.1, 0.2, 0.3, 0.4\}$. For each algorithm-environment pair, we identified the best performing exploration hyperparameter configuration (see Tables D.4–D.5).

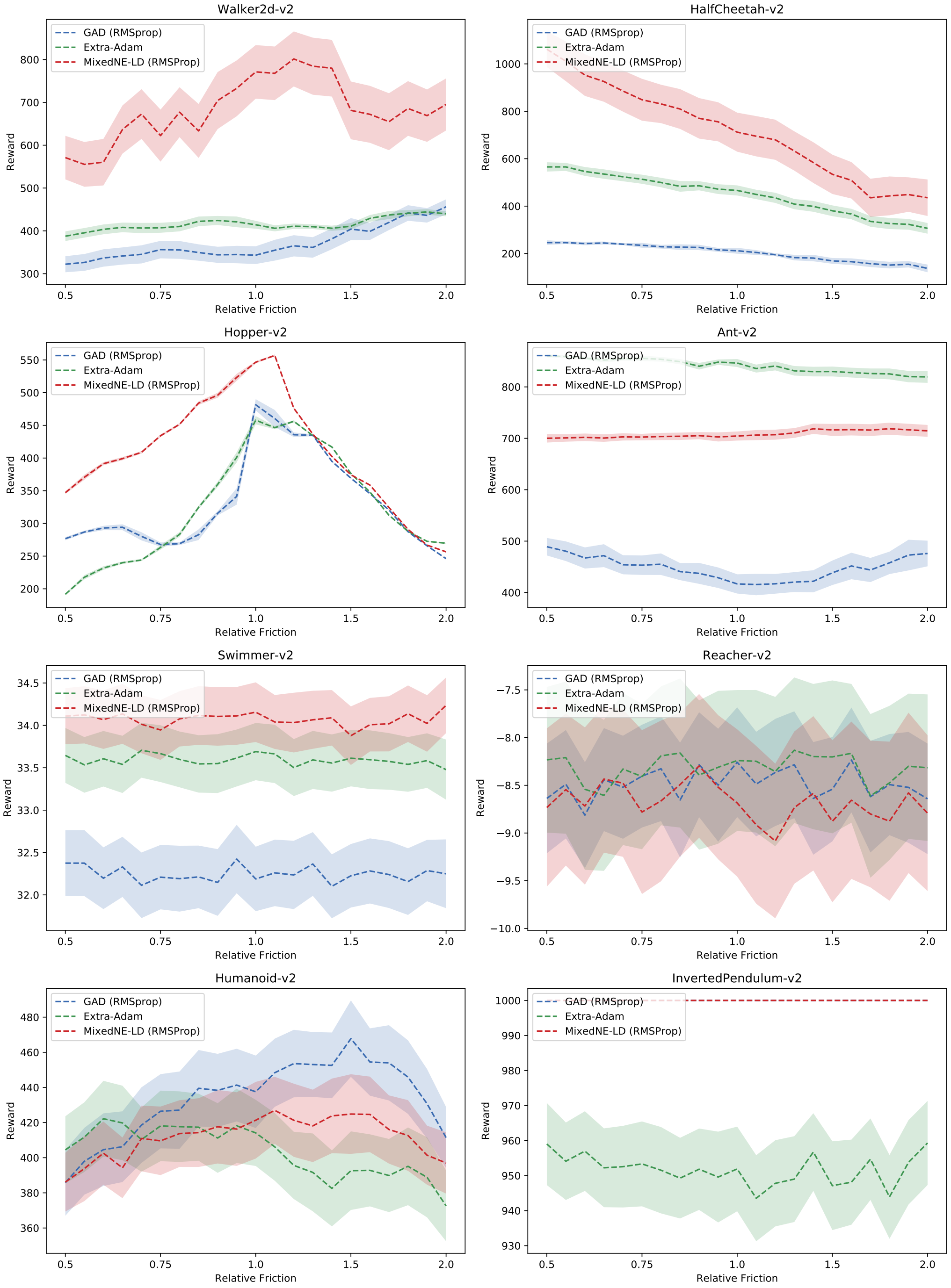


Figure 5.9: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed without adversarial perturbations, on a range of friction values not encountered during training.

Each algorithm is trained on 0.5M samples (i.e., 0.5M time steps in the environment). We run our experiments, for each environment, with 5 different seeds. The exploration noise is turned off for evaluation.

Evaluation: We evaluate the robustness of all the algorithms under different testing conditions, and in the presence of adversarial disturbances in the testing environment. We train the algorithms with the standard mass and friction variables in OpenAI Gym. At test time, we evaluate the learned policies by changing the mass and friction values (without adversarial perturbations) and estimating the cumulative rewards. As shown in Fig. 5.6 and Fig. 5.8, our Algorithm 14 outperforms the baselines Algorithm 15 (with GAD and Extra-Adam) in terms of robustness. Note that we obtain superior performance on the inverted pendulum, which is a failure case for [TEM19]. We also evaluate the robustness of the learned policies under both test condition changes, and adversarial disturbances (cf. Appendix D.6.1).

5.7.2 On-Policy (VPG) Experiments

In addition to the off-policy experiments, we test the effectiveness of the MixedNE-LD strategy with the vanilla policy gradient (VPG) method on a toy MDP problem. In particular, we design a two-player variant of VPG [SMSM00] algorithm (cf. Algorithm 16) by adapting the Algorithm 9.

Setup: We compare the performance of Algorithm 16 and Algorithm 17 (with GAD and Extra-Adam) on a parametrized class of MDPs $\left\{\mathcal{M}_\rho = \left(\mathcal{S}, \mathcal{A}, T_\rho, \gamma, P_0, R\right) : \rho \in [0, 0.4]\right\}$. Here $\mathcal{S} = [-10, 10]$, $\mathcal{A} = [-1, 1]$, and $R(s) = \sin(\sqrt{1.7}s) + \cos(\sqrt{0.3}s) + 3$. The transition dynamics T_ρ is defined as follows: given the current state and action (s_t, a_t), the next state is $s_{t+1} = s_t + a_t$ with probability $1 - \rho$, and $s_{t+1} = s_t + a'$ (where $a' \sim \mathrm{unif}([-1, 1])$) with probability ρ. We also ensure that $s_{t+1} \in [-10, 10]$.

For all the algorithms, we use a two-layer feedforward neural network structure of (16, 16, relu) for both actors (agent and adversary). The relevant hyperparameters are given in Tables D.6–D.8. Each algorithm is trained for 5000 steps. We run our experiments with 5 different seeds.

Evaluation: We train the algorithms with a nominal environment parameter $\rho = 0.2$, and evaluate the learned policies on a range of $\rho \in [0, 0.4]$ values. As shown in Fig. D.19 (cf. Appendix D.6.2), our Algorithm 16 outperforms the baselines Algorithm 17 (with GAD and Extra-Adam) in terms of robustness (in both two-player and one-player settings).

6 Conclusion and future work

6.1 Summary of the thesis

For the three fundamental tasks introduced in Chapter 1, we have shown that many important *non-convex* instances admit elegant solutions. Specifically, in the thesis, we have achieved the following:

- In Chapter 3, we have resolved a decades-old open problem, the *Latent Dirichlet Allocation* (LDA), by providing the first efficient and provably convergent sampling scheme for the non-convex Dirichlet distribution. Our approach relies on a classical idea of *mirror descent* (MD) from the optimization theory. By combining Langevin dynamics and MD, our theory revealed that the Dirichlet distributions are in fact *convex* distributions in disguise. We further developed analytical tools for converting convergence guarantees between the primal and the dual distributions, akin to the optimization theory of MD. On real datasets, our algorithm achieves the state-of-the-art.
- For min-max games, we developed a framework for studying the long-term behavior of training algorithms in Chapter 4. For any algorithm that can be cast as an generalized *Robbins–Monro* (RM) scheme, a generic template that subsumes most of the existing algorithms, our theory dictates the asymptotic convergence to the *internally chain-transitive* (ICT) sets of the *mean dynamics* (MD). Using this connection, we provided negative examples showing that state-of-the-art algorithms for min-max games suffer from convergence to spurious sets which are no way min-max optimal, thus theoretically corroborating the formidable difficulty of training generative adversarial networks and robust reinforcement learning.
- Finally, in Chapter 5, we showed how min-max games can be lifted to infinite dimension and solved using the classical *prox* techniques in optimization theory. Our approach leverages an intimate connection between the infinite-dimensional iterates and first-order sampling, which gives rise to practically efficient schemes. Our experiments demonstrated that our approach consistently outperforms state-of-the-arts on real

applications.

6.2 Future directions

This thesis left a number of important questions open, which we plan to investigate in the future:

6.2.1 Mirrored Langevin dynamics and non-convex sampling

The Dirichlet distribution is but one of the many examples of constrained and non-convex distributions. Our theory of mirrored Langevin dynamics suggests that a highly symmetric constraint, which is typically the case in real world applications, often hides convexity that is implicit in the primal form. Akin to the Latent Dirichlet Allocation, we expect mirrored Langevin dynamics to be widely applicable for these distributions by transforming them into the dual formulation, for which the non-convexity disappears (or at least alleviated).

6.2.2 Theoretical guarantees of Langevin dynamics for min-max games

Our approach in Chapter 5 exploits the *optimization* perspective of Langevin dynamics, i.e., how Langevin dynamics serves as a subsolver for iterates of an infinite-dimensional *optimization* algorithm. However, many natural questions remained untouched by our perspective:

1. From the *sampling* perspective, does the *stationary distribution* of Langevin dynamics always exist?
2. If yes, is the stationary distribution unique?
3. Does the stationary distribution behave similar to the ICT sets of (MD) (say, concentrating around these ICT sets)?

From a high-level point of view, these are the sampling counterparts of the questions that we answered in Chapter 4 for min-max games. As Langevin dynamics often serves as a bridge between optimization and sampling, we expect the above questions to provide valuable insights into solving min-max games, in a fashion that is similar to how Langevin dynamics characterizes non-convex optimization [ZLC17, RRT17].

6.2.3 Limit cycles in min-max games

Our negative examples Examples 4.5.1–4.5.3 raise an immediate question: how can we eliminate *limit cycles* in min-max games using optimization techniques?

One natural idea would be to modify the driving vector fields of the min-max optimization algorithm. For instance, [BRM$^+$18] considered decomposing the gradient fields into a "descending" (potential) and a "rotating" (Hamiltonian) component, and our theory in Chapter 4 implies that the limit cycles can arise in a complicated manner due to the Hamiltonian component. Therefore, it is reasonable to first "single out" the Hamiltonian component (or at least approximately) of the gradient fields of min-max games.

Physical theory suggests that evaluating the Hamiltonian component at a point amounts to a *contour integration.* We expect this new ingredient to play a key role in the future development of min-max optimization algorithms, as the Hamiltonian component is precisely the culprit for cycling.

A Appendix for Chapter 2

A.1 Equivalence formulations of optimistic mirror descent

In this appendix, we show that the $\boldsymbol{x}_t$ iterates in (2.3) of the main text is equivalent to the following iterates given in [CYL⁺12, RS13a]:

$$\begin{cases} \boldsymbol{x}_t & = \mathrm{MD}_\eta\left(\tilde{\boldsymbol{x}}_t, -A\boldsymbol{y}_{t-1}\right) \\ \tilde{\boldsymbol{x}}_{t+1} & = \mathrm{MD}_\eta\left(\tilde{\boldsymbol{x}}_t, -A\boldsymbol{y}_t\right) \end{cases}. \tag{A.1}$$

By the optimality condition for (A.1), we have

$$\nabla h(\boldsymbol{x}_t) = \nabla h(\tilde{\boldsymbol{x}}_t) - \eta\left(-A\boldsymbol{y}_{t-1}\right), \tag{A.2}$$

$$\nabla h(\tilde{\boldsymbol{x}}_t) = \nabla h(\tilde{\boldsymbol{x}}_{t-1}) - \eta\left(-A\boldsymbol{y}_{t-1}\right), \tag{A.3}$$

$$\nabla h(\tilde{\boldsymbol{x}}_{t-1}) = \nabla h(\boldsymbol{x}_{t-1}) + \eta\left(-A\boldsymbol{y}_{t-2}\right). \tag{A.4}$$

We hence get (2.3) by applying (A.4) to (A.3) and then (A.3) to (A.2).

A.2 Optimistic mirror descent

In this appendix, we prove Theorem 2.2, restated below for convenience.

Theorem A.1. *Suppose two players of a zero-sum game have played T rounds according to Algorithms 1–2 with $\eta = \frac{1}{2|A|_{\max}}$. Then*

1. *The $\boldsymbol{x}$-player suffers a $O\left(\frac{\log(T)}{T}\right)$ regret:*

$$\max_{\boldsymbol{z}\in\Delta_m} \sum_{t=3}^{T} \langle \boldsymbol{z}_t - \boldsymbol{z}, -A\boldsymbol{w}_t\rangle \le \Big(\log(T-2)+1\Big)\Big(20+\log m+\log n\Big)|A|_{\max} \tag{A.5}$$
$$= O\big(\log T\big)$$

and similarly for the $\boldsymbol{y}$-player.

2. *The strategies $(\boldsymbol{z}_T, \boldsymbol{w}_T)$ constitutes an $O\left(\frac{1}{T}\right)$-approximate equilibrium to the value of the game:*

$$|V - \langle \boldsymbol{z}_T, A\boldsymbol{w}_T\rangle| \le \frac{\left(20 + \log m + \log n\right)|A|_{\max}}{T-2} = O\left(\frac{1}{T}\right). \tag{A.6}$$

Proof. Define $\boldsymbol{x}^*$ as

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\Delta_m} \left\langle \boldsymbol{x}, -A\left(\frac{1}{T-2}\sum_{t=3}^{T} \boldsymbol{y}_t\right)\right\rangle. \tag{A.7}$$

We define an auxiliary individual regret $\mathrm{R}_T^{\boldsymbol{x}}$ as

$$\mathrm{R}_T^{\boldsymbol{x}} := \sum_{t=3}^{T} \langle \boldsymbol{x}_t - \boldsymbol{x}^*, -A\boldsymbol{y}_t\rangle. \tag{A.8}$$

Notice that this is the regret on the $\boldsymbol{x}_t$ sequence versus $\boldsymbol{y}_t$ sequence, while we are playing $\boldsymbol{z}_t$'s and $\boldsymbol{w}_t$'s in the algorithm.

We then have

$$\begin{aligned}
\mathrm{R}_T^{\boldsymbol{x}} &= \sum_{t=3}^{T} \langle \boldsymbol{x}_t - \boldsymbol{x}^*, -A\boldsymbol{y}_t\rangle \\
&= \langle \boldsymbol{x}_3 - \boldsymbol{x}^*, -A\boldsymbol{y}_3\rangle + \sum_{t=4}^{T} \langle \boldsymbol{x}_t - \boldsymbol{x}^*, -A\boldsymbol{y}_t\rangle \\
&\le 2|A|_{\max} + \sum_{t=4}^{T} \langle \boldsymbol{x}_t - \boldsymbol{x}^*, -A\boldsymbol{y}_t - \boldsymbol{g}_{t-1}\rangle + \sum_{t=4}^{T} \langle \boldsymbol{x}_t - \boldsymbol{x}^*, \boldsymbol{g}_{t-1}\rangle
\end{aligned}$$

where $\boldsymbol{g}_t := -2(t-2)A\boldsymbol{w}_t + 3(t-3)A\boldsymbol{w}_{t-1} - (t-4)A\boldsymbol{w}_{t-2}$. Inserting $\boldsymbol{w}_t = \frac{1}{t-2}\sum_{i=3}^{t} \boldsymbol{y}_i$ into the definition of $\boldsymbol{g}_t$, we get $\boldsymbol{g}_t = -2A\boldsymbol{y}_t + A\boldsymbol{y}_{t-1}$. Straightforward calculation then shows:

$$
\begin{aligned}
\mathrm{R}_T^{\boldsymbol{x}} &\leq 2|A|_{\max} + \sum_{t=4}^{T} \langle \boldsymbol{x}_t - \boldsymbol{x}^*, -A\boldsymbol{y}_t + 2A\boldsymbol{y}_{t-1} - A\boldsymbol{y}_{t-2}\rangle + \sum_{t=4}^{T} \langle \boldsymbol{x}_t - \boldsymbol{x}^*, -2A\boldsymbol{y}_{t-1} + A\boldsymbol{y}_{t-2}\rangle \\
&= 2|A|_{\max} + \sum_{t=4}^{T} \langle \boldsymbol{x}_t - \boldsymbol{x}^*, (-A\boldsymbol{y}_t + A\boldsymbol{y}_{t-1}) - (-A\boldsymbol{y}_{t-1} + A\boldsymbol{y}_{t-2})\rangle \\
&\qquad\qquad + \frac{1}{\eta} \sum_{t=4}^{T} \Big(D(\boldsymbol{x}^*, \boldsymbol{x}_{t-1}) - D(\boldsymbol{x}^*, \boldsymbol{x}_t) - D(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) \Big) \\
&= 2|A|_{\max} + \sum_{t=4}^{T-1} \langle \boldsymbol{x}_t - \boldsymbol{x}_{t+1}, -A\boldsymbol{y}_t + A\boldsymbol{y}_{t-1}\rangle + \langle \boldsymbol{x}_4 - \boldsymbol{x}^*, A\boldsymbol{y}_3 - A\boldsymbol{y}_2\rangle \\
&\quad + \langle \boldsymbol{x}_T - \boldsymbol{x}^*, -A\boldsymbol{y}_T + A\boldsymbol{y}_{T-1}\rangle + \frac{1}{\eta} \sum_{t=4}^{T} \Big(D(\boldsymbol{x}^*, \boldsymbol{x}_{t-1}) - D(\boldsymbol{x}^*, \boldsymbol{x}_t) - D(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) \Big) \\
&\leq 10|A|_{\max} + \sum_{t=4}^{T-1} \langle \boldsymbol{x}_t - \boldsymbol{x}_{t+1}, -A\boldsymbol{y}_t + A\boldsymbol{y}_{t-1}\rangle \\
&\qquad\qquad + \frac{1}{\eta} \sum_{t=4}^{T} \Big(D(\boldsymbol{x}^*, \boldsymbol{x}_{t-1}) - D(\boldsymbol{x}^*, \boldsymbol{x}_t) - D(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) \Big) \\
&\leq 10|A|_{\max} + \sum_{t=4}^{T-1} \|\boldsymbol{x}_t - x_{t+1}\|_1 \cdot |A|_{\max} \cdot \|\boldsymbol{y}_t - \boldsymbol{y}_{t-1}\|_1 \\
&\qquad\qquad + \frac{1}{\eta} \Big(D(\boldsymbol{x}^*, \boldsymbol{x}_3) - D(\boldsymbol{x}^*, \boldsymbol{x}_T) \Big) + \sum_{t=4}^{T} \frac{-1}{\eta} D(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) \\
&\leq 10|A|_{\max} + \frac{1}{2} \sum_{t=4}^{T-1} \Big(|A|_{\max} \cdot \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|_1^2 + |A|_{\max} \cdot \|\boldsymbol{y}_t - \boldsymbol{y}_{t-1}\|_1^2 \Big) \\
&\qquad\qquad + \frac{1}{\eta} \Big(D(\boldsymbol{x}^*, \boldsymbol{x}_3) - D(\boldsymbol{x}^*, \boldsymbol{x}_T) \Big) + \sum_{t=4}^{T} \frac{-1}{\eta} D(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}).
\end{aligned}
$$

Using the fact that h is 1-strongly convex with respect to the ℓ_1-norm, we have $-D(\boldsymbol{x}, \boldsymbol{x}') \leq -\frac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}'\|_1^2 \leq 0$. Also, we have $D(\boldsymbol{x}^*, \boldsymbol{x}_3) \leq \log m$. Combining these facts in the last inequality gives:

$$
\begin{aligned}
\mathrm{R}_T^{\boldsymbol{x}} &\leq 10|A|_{\max} + \frac{\log m}{\eta} + \frac{|A|_{\max}}{2} \sum_{t=4}^{T-1} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|_1^2 \\
&\qquad + \frac{|A|_{\max}}{2} \sum_{t=4}^{T-1} \|\boldsymbol{y}_t - \boldsymbol{y}_{t-1}\|_1^2 - \frac{1}{2\eta} \sum_{t=4}^{T} \|\boldsymbol{x}_{t-1} - \boldsymbol{x}_t\|_1^2.
\end{aligned}
$$

Similarly, for the second player we define

$$
\mathrm{R}_T^{\boldsymbol{y}} := \sum_{t=3}^{T} \langle \boldsymbol{y}_t - \boldsymbol{y}^*, A^\top \boldsymbol{x}_t \rangle \tag{A.9}
$$

where $\boldsymbol{y}^* := \arg\min_{\boldsymbol{y}} \left\langle \boldsymbol{y}, A^\top \left(\frac{1}{T-2}\sum_{t=3}^{T} \boldsymbol{x}_t\right)\right\rangle$. We then have

$$\begin{aligned} \mathrm{R}_T^{\boldsymbol{y}} \le 10|A|_{\max} + \frac{\log n}{\eta} + \frac{|A|_{\max}}{2}\sum_{t=4}^{T-1}\|\boldsymbol{y}_t - \boldsymbol{y}_{t+1}\|_1^2 \\ + \frac{|A|_{\max}}{2}\sum_{t=4}^{T-1}\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_1^2 - \frac{1}{2\eta}\sum_{t=4}^{T}\|\boldsymbol{y}_{t-1} - \boldsymbol{y}_t\|_1^2. \end{aligned}$$

Setting $\eta = \frac{1}{2|A|_{\max}}$, we get

$$\mathrm{R}_T^{\boldsymbol{x}} + \mathrm{R}_T^{\boldsymbol{y}} \le \left(20 + \log m + \log n\right)|A|_{\max}. \tag{A.10}$$

Now, recalling that $\boldsymbol{z}_T = \frac{\sum_{t=3}^{T}\boldsymbol{x}_t}{T-2}$ and $\boldsymbol{w}_T = \frac{\sum_{t=3}^{T}\boldsymbol{y}_t}{T-2}$ and using the definition of $\mathrm{R}_T^{\boldsymbol{x}}$ and $\mathrm{R}_T^{\boldsymbol{y}}$, we get

$$\frac{1}{T-2}\left(\mathrm{R}_T^{\boldsymbol{x}} + \mathrm{R}_T^{\boldsymbol{y}}\right) = \max_{\boldsymbol{x}\in\Delta_m}\langle \boldsymbol{x}, A\boldsymbol{w}_T\rangle - \min_{\boldsymbol{y}\in\Delta_n}\left\langle \boldsymbol{z}_T, A\boldsymbol{y}\right\rangle. \tag{A.11}$$

Furthermore, by the definition of the value of the game, we have

$$\min_{\boldsymbol{y}\in\Delta_n}\left\langle \boldsymbol{z}_T, A\boldsymbol{y}\right\rangle \le V \le \max_{\boldsymbol{x}\in\Delta_m}\langle \boldsymbol{x}, A\boldsymbol{w}_T\rangle. \tag{A.12}$$

We also trivially have

$$\min_{\boldsymbol{y}\in\Delta_n}\left\langle \boldsymbol{z}_T, A\boldsymbol{y}\right\rangle \le \langle \boldsymbol{z}_T, A\boldsymbol{w}_T\rangle \le \max_{\boldsymbol{x}\in\Delta_m}\langle \boldsymbol{x}, A\boldsymbol{w}_T\rangle. \tag{A.13}$$

Combining (A.11) - (A.13) in (A.10) then establishes (2.5):

$$|V - \langle \boldsymbol{z}_T, A\boldsymbol{w}_T\rangle| \le \frac{\left(20 + \log m + \log n\right)|A|_{\max}}{T-2}.$$

We now turn to (2.4).

Let $\mathrm{R}_T^{\boldsymbol{z}} := \max_{\boldsymbol{z}\in\Delta_m}\sum_{t=3}^{T}\langle \boldsymbol{z}_t - \boldsymbol{z}, -A\boldsymbol{w}_t\rangle$ and let $\tilde{\mathrm{R}}_T^{\boldsymbol{z}} := \sum_{t=3}^{T}\left\langle \boldsymbol{z}_t - \boldsymbol{z}_t^*, -A\boldsymbol{w}_t\right\rangle$ where $\boldsymbol{z}_t^* = \arg\min_{\boldsymbol{z}\in\Delta_m}\langle \boldsymbol{z}, -A\boldsymbol{w}_t\rangle$. Evidently we have $\mathrm{R}_T^{\boldsymbol{z}} \le \tilde{\mathrm{R}}_T^{\boldsymbol{z}}$. Notice that (with $\boldsymbol{w}_t^*$ similarly defined)

$$\begin{aligned} \left\langle \boldsymbol{z}_t - \boldsymbol{z}_t^*, -A\boldsymbol{w}_t\right\rangle &= \left\langle \boldsymbol{z}_t^*, A\boldsymbol{w}_t\right\rangle - \langle \boldsymbol{z}_t, A\boldsymbol{w}_t\rangle \\ &\le \left\langle \boldsymbol{z}_t^*, A\boldsymbol{w}_t\right\rangle - \left\langle \boldsymbol{z}_t, A\boldsymbol{w}_t^*\right\rangle \\ &\le \frac{\left(20 + \log m + \log n\right)|A|_{\max}}{t-2} \end{aligned} \tag{A.14}$$

by (A.10) and (A.11). Using these inequalities, we get

$$\begin{aligned}\frac{1}{T-2}\mathrm{R}^{\boldsymbol{z}}_T \le \frac{1}{T-2}\tilde{\mathrm{R}}^{\boldsymbol{z}}_T &= \frac{1}{T-2}\sum_{t=3}^{T}\left\langle \boldsymbol{z}_t-\boldsymbol{z}^*_t, -A\boldsymbol{w}_t\right\rangle \\ &\le \frac{1}{T-2}\sum_{t=3}^{T}\frac{\Big(20+\log m+\log n\Big)|A|_{\max}}{t-2} \\ &\le \frac{\Big(\log(T-2)+1\Big)\Big(20+\log m+\log n\Big)|A|_{\max}}{T-2}\end{aligned}$$

which finishes the proof. ■

A.3 Robust optimistic mirror descent

In this appendix, we prove Theorem 2.3, repeated below for convenience.

Theorem A.2 ($O(\sqrt{T})$-Adversarial Regret). *Suppose that* $\|\nabla f_t\|_* \le G$ *for all* t. *Then playing* T *rounds of Algorithm 3 with* $\eta_t = \frac{1}{G\sqrt{t}}$ *against an arbitrary sequence of convex functions has the following guarantee on the regret:*

$$\begin{aligned}\max_{\boldsymbol{x}\in\Delta_m}\sum_{t=1}^{T}\left\langle \boldsymbol{x}_t-\boldsymbol{x}, \nabla f_t(\boldsymbol{x}_t)\right\rangle &\le G\sqrt{T}\left(18+2D^2\right)+GD\Big(3\sqrt{2}+4D\Big) \\ &= O\Big(\sqrt{T}\Big).\end{aligned}$$

Proof. Define $\mathrm{R}^{\boldsymbol{x}}_T := \sum_{t=1}^{T}\left\langle \boldsymbol{x}_t-\boldsymbol{x}^*, \nabla f_t(\boldsymbol{x}_t)\right\rangle$ where $\boldsymbol{x}^* := \arg\min_{\boldsymbol{x}\in\Delta_m}\left\langle \boldsymbol{x}, \sum_{t=1}^{T}\nabla f_t(\boldsymbol{x}_t)\right\rangle$. Let $\tilde{\nabla}_t = 2\nabla f_t(\boldsymbol{x}_t) - \nabla f_{t-1}(\boldsymbol{x}_{t-1})$, and let $\eta_t = \frac{1}{\alpha\sqrt{t}}$ for some $\alpha > 0$ to be chosen later. Then

$$\begin{aligned}\mathrm{R}^{\boldsymbol{x}}_T &= \sum_{t=1}^{T}\langle \boldsymbol{x}_t-\boldsymbol{x}^*, \nabla f_t(\boldsymbol{x}_t)\rangle \\ &\le \sqrt{2}DG + \sum_{t=2}^{T}\langle \boldsymbol{x}_t-\boldsymbol{x}^*, \nabla f_t(\boldsymbol{x}_t)-\tilde{\nabla}_{t-1}\rangle + \sum_{t=2}^{T}\langle \boldsymbol{x}_t-\boldsymbol{x}^*, \tilde{\nabla}_{t-1}\rangle \\ &\le \sqrt{2}DG + \sum_{t=2}^{T}\langle \boldsymbol{x}_t-\boldsymbol{x}^*, \nabla f_t(\boldsymbol{x}_t)-\nabla f_{t-1}(\boldsymbol{x}_{t-1})\rangle \\ &\qquad\qquad - \sum_{t=2}^{T}\langle \boldsymbol{x}_t-\boldsymbol{x}^*, \nabla f_{t-1}(\boldsymbol{x}_{t-1})-\nabla f_{t-2}(\boldsymbol{x}_{t-2})\rangle + \sum_{t=2}^{T}\langle \boldsymbol{x}_t-\boldsymbol{x}^*, \tilde{\nabla}_{t-1}\rangle \\ &\le 3\sqrt{2}DG + \sum_{t=2}^{T-1}\langle \boldsymbol{x}_t-\boldsymbol{x}_{t+1}, \nabla f_t(\boldsymbol{x}_t)-\nabla f_{t-1}(\boldsymbol{x}_{t-1})\rangle + \sum_{t=2}^{T}\frac{1}{\eta_t}\Big(D(\boldsymbol{x}^*,\tilde{\boldsymbol{x}}_{t-1})-D(\boldsymbol{x}^*,\boldsymbol{x}_t)-D(\boldsymbol{x}_t,\tilde{\boldsymbol{x}}_{t-1})\Big) \\ &\le 3\sqrt{2}DG + \sum_{t=2}^{T-1}\left(\frac{\sqrt{t}G}{9}\|\boldsymbol{x}_t-\boldsymbol{x}_{t+1}\|^2+\frac{9G}{\sqrt{t}}\right) \\ &\qquad\qquad + \alpha\sum_{t=1}^{T}\sqrt{t}\Big(D(\boldsymbol{x}^*,\tilde{\boldsymbol{x}}_{t-1})-D(\boldsymbol{x}^*,\boldsymbol{x}_t)-D(\boldsymbol{x}_t,\tilde{\boldsymbol{x}}_{t-1})\Big).\end{aligned}$$

Using the joint convexity of $D(\boldsymbol{x}, \boldsymbol{y})$ in $\boldsymbol{x}$ and $\boldsymbol{y}$ and the strong convexity of the entropic mirror map, we get:

$$
\begin{aligned}
-D(\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_{t-1}) &\leq -\frac{1}{2}\|\tilde{\boldsymbol{x}}_t - \boldsymbol{x}_{t+1}\|^2 \\
&\leq -\frac{1}{4}\left\|\frac{t-1}{t}(\boldsymbol{x}_t - \boldsymbol{x}_{t+1})\right\|^2 + \frac{1}{2}\left(\frac{1}{t}\right)^2 \|\boldsymbol{x}_c - \boldsymbol{x}_{t+1}\|^2 \\
&\leq -\frac{(t-1)^2}{4t^2}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 + \frac{D^2}{t^2},
\end{aligned}
$$

and

$$
D(\boldsymbol{x}^*, \tilde{\boldsymbol{x}}_t) \leq \frac{t-1}{t} D(\boldsymbol{x}^*, \boldsymbol{x}_t) + \frac{1}{t} D\left(\boldsymbol{x}^*, \boldsymbol{x}_c\right).
$$

Meanwhile, straightforward calculations show that

$$
\sum_{t=2}^{T} \frac{D\left(\boldsymbol{x}^*, \boldsymbol{x}_c\right)}{\sqrt{t}} \leq 2D^2\sqrt{T},
$$

and

$$
\begin{aligned}
\sum_{t=2}^{T}\left(\sqrt{t}\cdot\frac{t-1}{t} D(\boldsymbol{x}^*, \boldsymbol{x}_{t-1}) - \sqrt{t} D(\boldsymbol{x}^*, \boldsymbol{x}_t)\right) &\leq \sum_{t=2}^{T}\left(\sqrt{t-1} D(\boldsymbol{x}^*, \boldsymbol{x}_{t-1}) - \sqrt{t} D(\boldsymbol{x}^*, \boldsymbol{x}_t)\right) \\
&\leq D(\boldsymbol{x}^*, \boldsymbol{x}_1) \leq D^2.
\end{aligned}
$$

We can hence continue as

$$
\begin{aligned}
\mathrm{R}_T^{\boldsymbol{x}} \leq{} & 3\sqrt{2}DG + \sum_{t=2}^{T-1}\left(\frac{\sqrt{t}}{9} G\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 + \frac{9G}{\sqrt{t}}\right) + 2\alpha D^2\sqrt{T} \\
& + \alpha D^2 - \frac{\alpha}{4}\sum_{t=2}^{T}\sqrt{t}\cdot\left(\frac{t-1}{t}\right)^2 \|\boldsymbol{x}_{t-1} - \boldsymbol{x}_t\|^2 + \alpha D^2 \sum_{t=2}^{T}\frac{\sqrt{t}}{t^2}.
\end{aligned}
\tag{A.15}
$$

Elementary calculations further show

$$
\begin{aligned}
&\sum_{t=2}^{T-1}\frac{9G}{\sqrt{t}} \leq 18G\sqrt{T}, \\
&\sum_{t=2}^{T}\frac{1}{t\sqrt{t}} \leq 3.
\end{aligned}
$$

Finally, since $(\frac{t-1}{t})^2 \geq \frac{4}{9}$ for $t \geq 3$, we can further bound (A.15) as

$$
\begin{aligned}
\mathrm{R}_t^{\boldsymbol{x}} \leq{} & 3\sqrt{2}DG + 18G\sqrt{T} + 2\alpha D^2\sqrt{T} + 4\alpha D^2 \\
& + \left(\frac{G}{9}\sum_{t=2}^{T-1}\sqrt{t}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 - \frac{\alpha}{4}\cdot\frac{4}{9}\sum_{t=2}^{T-1}\sqrt{t+1}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2\right).
\end{aligned}
$$

The proof is finished by choosing $\alpha = G$. ■

B Appendix for Chapter 3

B.1 Proof of Theorem 3.2

We first focus on the convergence for total variation and relative entropy, since they are in fact quite trivial. The proof for the 2-Wasserstein distance requires a bit more work.

B.1.1 Total variation and relative entropy

Since h is strictly convex, ∇h is one-to-one, and hence

$$\begin{aligned} d_{\mathrm{TV}}(\nabla h\#\mu_1, \nabla h\#\mu_2) &= \frac{1}{2}\sup_E |\nabla h\#\mu_1(E) - \nabla h\#\mu_2(E)| \\ &= \frac{1}{2}\sup_E \left|\mu_1\left(\nabla h^{-1}(E)\right) - \mu_2\left(\nabla h^{-1}(E)\right)\right| \\ &= d_{\mathrm{TV}}(\mu_1, \mu_2). \end{aligned}$$

On the other hand, it is well-known that applying a one-to-one mapping to distributions leaves the relative entropy intact. Alternatively, we may also simply write (letting $\nu_i = \nabla h\#\mu_i$):

$$\begin{aligned} D(\nu_1 \| \nu_2) &= \int \log \frac{\mathrm{d}\nu_1}{\mathrm{d}\nu_2}\mathrm{d}\nu_1 \\ &= \int \log\left(\frac{\mathrm{d}\nu_1}{\mathrm{d}\nu_2} \circ \nabla h\right)\mathrm{d}\mu_1 && \text{by (B.5) below} \\ &= \int \log \frac{\mathrm{d}\mu_1}{\mathrm{d}\mu_2}\mathrm{d}\mu_1 && \text{by (3.2)} \\ &= D(\mu_1 \| \mu_2) \end{aligned}$$

The "in particular" part follows from noticing that $\boldsymbol{y}^t \sim \nabla h\#\boldsymbol{x}^t$ and $\boldsymbol{Y}_\infty \sim \nabla h\#\boldsymbol{X}_\infty$.

B.1.2 2-Wasserstein distance

Now, let h be ρ-strongly convex. The most important ingredient of the proof is Lemma B.1 below, which is conceptually clean. Unfortunately, for the sake of rigor, we must deal with certain intricate regularity issues in the Optimal Transport theory. If the reader wishes, she/he can simply assume that the quantities (B.1) and (B.2) below are well-defined, which is always satisfied by any practical mirror map, and skip all the technical part about the well-definedness proof.

For the moment, assume $h \in \mathcal{C}^5$; the general case is given at the end. Every convex h generates a Bregman divergence via $B_h(\boldsymbol{x}, \boldsymbol{x}') := h(\boldsymbol{x}) - h(\boldsymbol{x}') - \langle \nabla h(\boldsymbol{x}'), \boldsymbol{x} - \boldsymbol{x}' \rangle$. The following key lemma allows us to relate guarantees in $\mathcal{W}_2$ between $\boldsymbol{x}^t$'s and $\boldsymbol{y}^t$'s. It can be seen as a generalization of the classical duality relation (B.4) in the space of probability measures.

Lemma B.1 (Duality of Wasserstein Distances). *Let μ_1, μ_2 be probability measures satisfying Assumptions* (A3.2)–(A3.3). *If h is ρ-strongly convex and $\mathcal{C}^5$, then the* (B.1) *and* (B.2) *below are well-defined:*

$$\mathcal{W}_{B_h}(\mu_1, \mu_2) := \inf_{T: T\#\mu_1 = \mu_2} \int B_h\left(\boldsymbol{x}, T(\boldsymbol{x})\right) \mathrm{d}\mu_1(\boldsymbol{x}) \tag{B.1}$$

and (notice the exchange of inputs on the right-hand side)

$$\mathcal{W}_{B_h^\star}(\nu_1, \nu_2) := \inf_{T: T\#\nu_1 = \nu_2} \int B_h^\star\left(T(\boldsymbol{y}), \boldsymbol{y}\right) \mathrm{d}\nu_1(\boldsymbol{y}). \tag{B.2}$$

Furthermore, we have

$$\mathcal{W}_{B_h}(\mu_1, \mu_2) = \mathcal{W}_{B_h^\star}(\nabla h\#\mu_1, \nabla h\#\mu_2). \tag{B.3}$$

Before proving the lemma, let us see that the relation in $\mathcal{W}_2$ is a simple corollary of Lemma B.1. Since h is ρ-strongly convex, it is classical that, for any $\boldsymbol{x}$ and $\boldsymbol{x}'$,

$$\frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{x}'\|^2 \le B_h(\boldsymbol{x}, \boldsymbol{x}') = B_h^\star(\nabla h(\boldsymbol{x}'), \nabla h(\boldsymbol{x})) \le \frac{1}{2\rho}\|\nabla h(\boldsymbol{x}) - \nabla h(\boldsymbol{x}')\|^2. \tag{B.4}$$

Using Lemma B.1 and the fact that $\boldsymbol{y}^t \sim \nabla h\#\boldsymbol{x}^t$ and $\boldsymbol{Y}_\infty \sim \nabla h\#\boldsymbol{X}_\infty$, we conclude $\mathcal{W}_2(\boldsymbol{x}^t, \boldsymbol{X}_\infty) \le \frac{1}{\rho}\mathcal{W}_2(\boldsymbol{y}^t, \boldsymbol{X}_\infty)$. It hence remains to prove Lemma B.1 when $h \in \mathcal{C}^5$.

Proof of Lemma B.1 when $h \in \mathcal{C}^5$

We first prove that (B.2) is well-defined by verifying the sufficient conditions in Theorem 3.6 of [DPF14]. Specifically, we will verify **(C0)**-**(C2)** in p.554 of [DPF14] when the transport cost is $B_h^\star$.

Since h is ρ-strongly convex, ∇h is injective, and hence $\nabla h^\star = (\nabla h)^{-1}$ is also injective, which implies that $h^\star$ is strictly convex. On the other hand, the strong convexity of h implies $\nabla^2 h^\star \preceq \frac{1}{\rho} I$, and hence $B^\star_h$ is globally upper bounded by a quadratic function.

We now show that the conditions **(C0)**-**(C2)** are satisfied. Since we have assumed $h \in \mathcal{C}^5$, we have $B^\star_h \in \mathcal{C}^4$. Since $B^\star_h$ is upper bounded by a quadratic function, the condition **(C0)** is trivially satisfied. On the other hand, since $h^\star$ is strictly convex, simple calculation reveals that, for any $\boldsymbol{y}'$, the mapping $\boldsymbol{y} \to \nabla_{\boldsymbol{y}'} B_{h^\star}(\boldsymbol{y}, \boldsymbol{y}')$ is injective, which is **(C1)**. Similarly, for any $\boldsymbol{y}$, the mapping $\boldsymbol{y}' \to \nabla_{\boldsymbol{y}} B_{h^\star}(\boldsymbol{y}, \boldsymbol{y}')$ is also injective, which is **(C2)**. By **Theorem 3.6** in [DPF14], (B.2) is well-defined.

We now turn to (B.3), which will automatically establish the well-definedness of (B.1). We first need the following equivalent characterization of $\nabla h \# \mu = \nu$ [Vil08]:

$$\int f \mathrm{d}\nu = \int f \circ \nabla h \mathrm{d}\mu \tag{B.5}$$

for all measurable f. Using (B.5) in the definition of $\mathcal{W}_{B^\star_h}$, we get

$$\begin{aligned}\mathcal{W}_{B^\star_h}(\nabla h \# \mu_1, \nabla h \# \mu_2) &= \inf_T \int B^\star_h\left(T(\boldsymbol{y}), \boldsymbol{y}\right) \mathrm{d}\nabla h \# \mu_1(\boldsymbol{y}) \\ &= \inf_T \int B^\star_h\Big((T \circ \nabla h)(\boldsymbol{x}), \nabla h(\boldsymbol{x})\Big) \mathrm{d}\mu_1(\boldsymbol{x}),\end{aligned}$$

where the infimum is over all T such that $T\#(\nabla h \# \mu_1) = \nabla h \# \mu_2$. Using the classical duality $B_h(\boldsymbol{x}, \boldsymbol{x}') = B^\star_h(\nabla h(\boldsymbol{x}'), \nabla h(\boldsymbol{x}))$ and $\nabla h \circ \nabla h^\star(\boldsymbol{x}) = \boldsymbol{x}$, we may further write

$$\mathcal{W}_{B^\star_h}(\nabla h \# \mu_1, \nabla h \# \mu_2) = \inf_T \int B_h\Big(\boldsymbol{x}, (\nabla h^\star \circ T \circ \nabla h)(\boldsymbol{x})\Big) \mathrm{d}\mu_1(\boldsymbol{x}) \tag{B.6}$$

where the infimum is again over all T such that $T\#(\nabla h \# \mu_1) = \nabla h \# \mu_2$. In view of (B.6), the proof would be complete if we can show that $T\#(\nabla h \# \mu_1) = \nabla h \# \mu_2$ if and only if $(\nabla h^\star \circ T \circ \nabla h)\#\mu_1 = \mu_2$.

For any two maps T_1 and T_2, we claim that

$$(T_1 \circ T_2)\#\mu = T_1\#\left(T_2\#\mu\right). \tag{B.7}$$

Indeed, for any Borel set E, we have, by definition of the push-forward,

$$\begin{aligned}(T_1 \circ T_2)\#\mu(E) &= \mu\big((T_1 \circ T_2)^{-1}(E)\big) \\ &= \mu\big((T_2^{-1} \circ T_1^{-1})(E)\big).\end{aligned}$$

On the other hand, recursively applying the definition of push-forward to $T_1\#\left(T_2\#\mu\right)$ gives

$$\begin{aligned}T_1\#\left(T_2\#\mu\right)(E) &= T_2\#\mu\big(T^{-1}(E)\big) \\ &= \mu\big((T_2^{-1} \circ T_1^{-1})(E)\big)\end{aligned}$$

which establishes (B.7).

Assume that $T\#(\nabla h\#\mu_1) = \nabla h\#\mu_2$. Then we have

$$\begin{aligned}
(\nabla h^\star \circ T \circ \nabla h)\#\mu_1 &= \nabla h^\star\#(T\#(\nabla h\#\mu_1)) && \text{by (B.7)}\\
&= \nabla h^\star\#(\nabla h\#\mu_2) && \text{since } T\#(\nabla h\#\mu_1) = \nabla h\#\mu_2\\
&= (\nabla h^\star \circ \nabla h)\#\mu_2 && \text{by (B.7) again}\\
&= \mu_2.
\end{aligned}$$

On the other hand, if $(\nabla h^\star \circ T \circ \nabla h)\#\mu_1 = \mu_2$, then composing both sides by ∇h and using (B.7) yields $T\#(\nabla h\#\mu_1) = \nabla h\#\mu_2$, which finishes the proof.

When h is only $\mathcal{C}^2$

When h is only $\mathcal{C}^2$, we will directly resort to (B.4). Let T be any map such that $T\#(\nabla h\#\mu_1) = \nabla h\#\mu_2$, and consider the optimal transportation problem $\inf_T \int \|\boldsymbol{y} - T(\boldsymbol{y})\|^2 \mathrm{d}\nabla h\#\mu_1(\boldsymbol{y})$. By (B.4) and (B.5), we have

$$\begin{aligned}
\inf_T \int \|\boldsymbol{y} - T(\boldsymbol{y})\|^2 \mathrm{d}\nabla h\#\mu_1(\boldsymbol{y}) &= \inf_T \int \|\nabla h(\boldsymbol{x}) - (T \circ \nabla h)(\boldsymbol{x}))\|^2 \mathrm{d}\mu_1(\boldsymbol{x})\\
&\geq \rho^2 \inf_T \int \|\boldsymbol{x} - (\nabla h^\star \circ T \circ \nabla h)(\boldsymbol{x}))\|^2 \mathrm{d}\mu_1(\boldsymbol{x})
\end{aligned}$$

where the infimum is over all T such that $T\#(\nabla h\#\mu_1) = \nabla h\#\mu_2$. But as shown in Appendix B.1.2, this is equivalent to $(\nabla h^\star \circ T \circ \nabla h)\#\mu_1 = \mu_2$. The proof is finished by noting $\boldsymbol{y}^t \sim \nabla h\#\boldsymbol{x}^t$ and $\boldsymbol{Y}_\infty \sim \nabla h\#\boldsymbol{X}_\infty$.

B.2 Proof of Lemma 3.1

Straightforward calculations in convex analysis shows

$$\begin{aligned}
&\frac{\partial h}{\partial x_i} = \log\frac{x_i}{x_{d+1}}, \qquad \frac{\partial^2 h}{\partial x_i \partial x_j} = \delta_{ij} x_i^{-1} + x_{d+1}^{-1},\\
&h^\star(\boldsymbol{y}) = \log\left(1 + \sum_{i=1}^d e^{y_i}\right), \quad \frac{\partial h^\star}{\partial y_i} = \frac{e^{y_i}}{1 + \sum_{i=1}^d e^{y_i}},
\end{aligned} \tag{B.8}$$

which proves that h is 1-strongly convex.

Let $\mu = e^{-V(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}$ be the target distribution and define $\nu = e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y} := \nabla h\#\mu$. By (3.2), we have

$$W \circ \nabla h = V + \log\det\nabla^2 h. \tag{B.9}$$

Since $\nabla^2 h(\boldsymbol{x}) = \mathrm{diag}[x_i^{-1}] + x_{d+1}^{-1}\mathbb{1}\mathbb{1}^\top$ where $\mathbb{1}$ is the all 1 vector, the well-known matrix determinant lemma "$\det(A+\mathbf{u}\mathbf{v}^\top) = (1+\mathbf{v}^\top A^{-1}\mathbf{u})\det A$" gives

$$\begin{aligned}\log\det\nabla^2 h(\boldsymbol{x}) &= \log\left(1+x_{d+1}^{-1}\sum_{i=1}^d x_i\right)\cdot\prod_{i=1}^d x_i^{-1}\\ &= -\sum_{i=1}^{d+1}\log x_i = -\sum_{i=1}^d \log x_i - \log\left(1-\sum_{i=1}^d x_i\right).\end{aligned} \tag{B.10}$$

Composing both sides of (B.9) with $\nabla h^\star$ and using (B.8), (B.10), we then finish the proof by computing

$$\begin{aligned}W(\boldsymbol{y}) &= V\circ\nabla h^\star(\boldsymbol{y}) - \sum_{i=1}^d y_i + (d+1)\log\left(1+\sum_{i=1}^d e^{y_i}\right)\\ &= V\circ\nabla h^\star(\boldsymbol{y}) - \sum_{i=1}^d y_i + (d+1)h^\star(\boldsymbol{y}).\end{aligned}$$

B.3 Proof of Lemma 3.2

The proof relies on rather straightforward computations.

1. In order to show $e^{-(W+C)} = \nabla h \# e^{-V}$ for some constant C, we will verify the Monge-Ampère equation:

   $$e^{-V} = e^{-(W\circ\nabla h + C)}\det\nabla^2 h \tag{B.11}$$

   for $V=\sum_{i=1}^N V_i$ and $W=\sum_{i=1}^N W_i$, where W_i is defined via (3.14). By (3.14), it holds that

   $$\frac{1}{C_i}e^{-NV_i} = e^{-NW_i\circ\nabla h}\det\nabla^2 h,\quad C_i := \frac{1}{\int e^{-NV_i}}. \tag{B.12}$$

   Multiplying (B.12) for $i=1,2,\dots,N$, we get

   $$\prod_{i=1}^N \frac{1}{C_i}e^{-NV} = e^{-NW\circ\nabla h}\left(\det\nabla^2 h\right)^N. \tag{B.13}$$

   The first claim follows by taking the N^{th} root of (B.13).

2. The second claim directly follows by (B.12).

3. Trivial.

4. By (B.11) and (B.12) and using $\nabla h^\star\circ\nabla h(\boldsymbol{x}) = \boldsymbol{x}$, we get

   $$W_i = V_i\circ\nabla h^\star + \frac{1}{N}\log\det\nabla^2 h(\nabla h^\star) - \log C_i, \tag{B.14}$$

   $$W = V\circ\nabla h^\star + \log\det\nabla^2 h(\nabla h^\star) - C, \tag{B.15}$$

which implies $N\nabla W_i - \nabla W = \nabla^2 h^\star \left(N\nabla V_i \circ \nabla h^\star - \nabla V \circ \nabla h^\star\right)$. Since h is 1-strongly convex, $h^\star$ is 1-Lipschitz gradient, and therefore the spectral norm of $\nabla^2 h^\star$ is upper bounded by 1. In the case of $b = 1$, the final claim follows by noticing

$$\begin{aligned}
\mathbb{E}\|\tilde{\nabla} W - \nabla W\|^2 &= \frac{1}{N}\sum_{i=1}^{N} \|N\nabla W_i - \nabla W\|^2 && \text{(B.16)}\\
&= \frac{1}{N}\sum_{i=1}^{N} \|\nabla^2 h^\star \left(N\nabla V_i \circ \nabla h^\star - \nabla V \circ \nabla h^\star\right)\|^2 && \text{(B.17)}\\
&\le \frac{\|\nabla^2 h^\star\|_{\text{spec}}^2}{N}\sum_{i=1}^{N} \|N\nabla V_i \circ \nabla h^\star - \nabla V \circ \nabla h^\star\|^2 && \text{(B.18)}\\
&\le \mathbb{E}\|\tilde{\nabla} V - \nabla V\|^2. && \text{(B.19)}
\end{aligned}$$

The proof for general batch-size b is exactly the same, albeit with more cumbersome notation.

B.4 Proof of Theorem 3.3

The proof is a simple combination of the existing result in [DMM18] and our theory in Section 3.3.

By Theorem 3.2, we only need to prove that the inequality (3.15) holds for $D(\tilde{\boldsymbol{y}}^T \| e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y})$, where $\tilde{\boldsymbol{y}}^T$ is to be defined below. By assumption, W is unconstrained and satisfies $LI \succeq \nabla^2 W \succeq 0$. By Lemma 3.2, the stochastic gradient $\tilde{\nabla} W$ is unbiased and satisfies

$$\mathbb{E}\|\tilde{\nabla} W - \nabla W\|^2 \le \mathbb{E}\|\tilde{\nabla} V - \nabla V\|^2 = \sigma^2.$$

Pick a random index[1] $t \in \{1, 2, \ldots, T\}$ and set $\tilde{\boldsymbol{y}}^T := \boldsymbol{y}^t$. Then Corollary 18 of [DMM18] with $D^2 = \sigma^2$ and $M_2 = 0$ implies $D(\tilde{\boldsymbol{y}}^T \| e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y}) \le \epsilon$, provided

$$\beta \le \min\left\{\frac{\epsilon}{2\left(Ld + \sigma^2\right)}, \frac{1}{L}\right\}, \quad T \ge \frac{\mathcal{W}_2^2(\boldsymbol{y}^0, e^{-W(\boldsymbol{y})}\mathrm{d}\boldsymbol{y})}{\beta\epsilon}. \tag{B.20}$$

Solving for T in terms of ϵ establishes the theorem.

B.5 Stochastic gradients for Dirichlet posteriors

In order to apply SMLD, one must have, for each term V_i, the corresponding dual W_i defined via (3.14). In this appendix, we derive a closed-form expression in the case of the Dirichlet posterior (3.10).

---

[1] The analysis in [DMM18] provides guarantees on the probability measure $\nu_T := \frac{1}{N}\sum_{t=1}^{T} \nu_t$ where $\boldsymbol{y}^t \sim \nu_t$. The $\tilde{\boldsymbol{y}}^T$ defined here has law ν_T.

Recall that the Dirichlet posterior (3.10) consists of a Dirichlet prior and categorical data observations [FKG10]. Let $N := \sum_{\ell=1}^{d+1} n_\ell$, where n_ℓ is the number of observations for category ℓ, and suppose that the parameters α_ℓ's are given. If the i^{th} data is in category $c_i \in \{1, 2, \dots, d+1\}$, then we can define $V_i(\boldsymbol{x}) := -\sum_{\ell=1}^{d+1} \mathbb{1}_{\{\ell=c_i\}} \log x_\ell - \frac{1}{N}\sum_{\ell=1}^{d+1} (\alpha_\ell - 1)\log x_\ell$ so that Assumption (3.12) holds. In view of Lemma 3.1, The corresponding dual W_i is, up to a constant, given by

$$W_i(\boldsymbol{y}) = -\sum_{\ell=1}^{d} \mathbb{1}_{\{\ell=c_i\}} y_\ell - \sum_{\ell=1}^{d} \frac{\alpha_\ell}{N} y_\ell + h^\star + \left(\sum_{\ell=1}^{d+1} \frac{\alpha_\ell}{N} \right) h^\star(\boldsymbol{y}). \tag{B.21}$$

Similarly, if we take a mini-batch B of the data with $|B| = b$, then

$$\frac{N}{b}\tilde{W}(\boldsymbol{y}) := \frac{N}{b}\sum_{i\in B} W_i(\boldsymbol{y}) = -\sum_{\ell=1}^{d}\left(\frac{Nm_\ell}{b} + \alpha_\ell\right) y_\ell + \left(N + \sum_{\ell=1}^{d+1}\alpha_\ell\right) h^\star(\boldsymbol{y}), \tag{B.22}$$

where m_ℓ is the number of observations of category ℓ in the set B. Apparently, the gradient of (B.22) is (3.16).

B.6 More on experiments

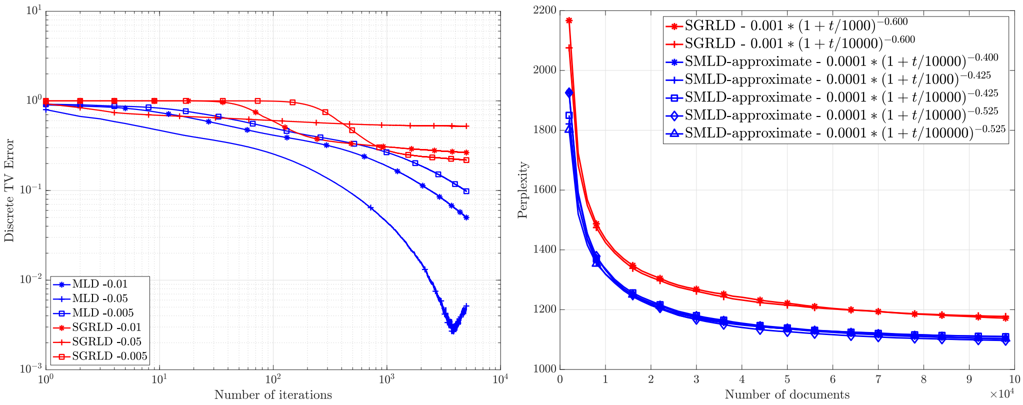
B.6.1 Synthetic Data

Fig. B.1(a) reports the total variation error along the 8^{th} dimension of the synthetic experiment in Section 3.5.1. Compared to Fig. 3.1(a) in the main text, it is evident that MLD achieves an even stronger performance than SGRLD, especially in the saturation error phase.

B.6.2 Comparison against SGRHMC for Latent Dirichlet Allocation

The only difference between the experimental setting of [MCF15] and the main text is the number of topics (50 vs. 100). In this appendix, we run SMLD-approximate under the setting of [MCF15] and directly compare against the results reported in [MCF15]. We have also included the SGRLD as a baseline.

Fig. B.1(b) reports the perplexity on the test data. According to [MCF15], the best perplexity achieved by SGRHMC up to 10000 documents is approximately 1400, which is worse than the 1323 by SMLD-approximate. Moreover, from Figure 3 of [MCF15], we see that the SGRHMC yields comparable performance as SGRLD for 2 out 3 independent runs, especially in the beginning phase, whereas the SMLD-approximate has sizable lead over SGRLD at any stage of the experiment. The potential reason for this improvement is, similar to SGRLD, that the SGRHMC exploits the Riemannian Hamiltonian dynamics, which is more complicated than MLD and hence more sensitive to the discretization error.



(a) Synthetic data, 8th dimension.

(b) LDA on Wikipedia corpus, 50 topics.

C Appendix for Chapter 4

C.1 Asymptotic pseudotrajectories

In this appendix, we discuss how the algorithms discussed in Section 4.3 fit within the general stochastic approximation framework of Section 4.4.2. Specifically, we prove the general conditions of Theorem 4.1 and Proposition 4.1 which guarantee that Algorithms 4.1–4.5 generate asymptotic pseudotrajectories of the mean dynamics (MD).

C.1.1 Generalities and preliminaries

Before doing so, we will require some background material on asymptotic pseudotrajectories. Following [BH96] and [Ben99], we first recall the definition of the "effective time" $\tau_n = \sum_{k=1}^{n} \gamma_k$ as the time that has elapsed at the n-th iteration of the discrete-time process Z_n; recall also the definition (4.6) of the continuous-time interpolation $Z(t)$ of Z_n as

$$Z(t) = Z_n + \frac{t - \tau_n}{\tau_{n+1} - \tau_n}(Z_{n+1} - Z_n) \tag{4.6}$$

We will further require the "continuous-to-discrete" correspondence

$$M(t) = \sup\{n \geq 1 : t \geq \tau_n\} \tag{C.1}$$

which measures the number of iterations required for the effective time τ_n of the process to reach the timestamp t; for future use, we also define the quantity

$$M_n \equiv M_n(T) = M(\tau_n + T). \tag{C.2}$$

Finally, given an arbitrary sequence A_n, we will denote its piecewise constant interpolation as

$$\overline{A}(t) = A_n \quad \text{for all } t \in [\tau_n, \tau_{n+1}],\ n \geq 1. \tag{C.3}$$

Using this notation, the (affinely) interpolated process $Z(t)$ can be expressed in integral form as

$$Z(t) = Z(0) + \int_0^t [V(\overline{Z}(s)) + \overline{W}(s)]\, ds \tag{C.4}$$

where W_n denotes the generalized error term of (RM).

With all this in hand, [Ben99, Prop. 4.1] provides the following general condition for $Z(t)$ to be an APT of the mean dynamics (4.7):

Proposition C.1. *Suppose that $Z(t)$ is bounded and satisfies the general condition*

$$\lim_{t\to\infty} \Delta(t;T) = 0 \quad \textit{for all } T > 0, \tag{C.5}$$

where

$$\Delta(t;T) = \sup\nolimits_{0\leq h\leq T} \left\| \textstyle\int_t^{t+h} \overline{W}(s)\, ds \right\|. \tag{C.6}$$

Then, $Z(t)$ is an APT of (MD).

C.1.2 Proof of Theorem 4.1

Our proof of Theorem 4.1 revolves around the direct verification of the requirement (C.5) of Proposition C.1 via the use of maximal inequalities and martingale limit theory.[1] For convenience, we restate the theorem below in full:

Theorem 4.1. *Suppose that* (RM) *is run with a step-size policy γ_n such that $\sum_n \gamma_n = \infty$*, $\lim_n \gamma_n = 0$, *and Assumptions* (A5.1)–(A5.2) *hold. Then, with probability* 1, *one of the following holds: a*) *Z_n is an APT of* (MD); *or b*) *Z_n is unbounded* (*and hence, non-convergent*).

Proof. Our proof relies on the Burkholder–Davis–Gundy (BDG) inequality [Bur73, HH80] which bounds the maximal value of a martingale S_n via its quadratic variation as

$$c_2\, \mathbb{E}\left[\sum_{k=1}^n (S_k - S_{k-1})^2\right] \leq \mathbb{E}\left[\max_{k=1,\dots,n} |S_k|^2\right] \leq C_2\, \mathbb{E}\left[\sum_{k=1}^n (S_k - S_{k-1})^2\right], \tag{BDG}$$

where $c_2, C_2 > 0$ are universal constants. As such, applying (BDG) to the martingale $S_m =$

---

[1] [Ben99] provides a set of sufficient conditions for (C.5) to hold when $Z(t)$ is generated by a RM scheme with $B_n = 0$ and $\sup_n \sigma_n < \infty$; however, our setting requires a more general treatment.

$\sum_{k=n}^{m} \gamma_k U_k$ (after an appropriate shift of the starting time), we get

$$\begin{aligned}\mathbb{E}\left[\sup_{n\le m\le M_n}\left\|\sum_{k=n}^{m}\gamma_k U_k\right\|^2\right] &\le C_2\,\mathbb{E}\left[\sum_{k=n}^{M_n}\gamma_k^2\|U_k\|^2\right]\\ &= C_2\sum_{k=n}^{M_n}\gamma_k^2\sigma_k^2 = C_2\int_{\tau_n}^{\tau_n+T}\overline{\gamma}^2(s)\overline{\sigma}^2(s)\,ds,\end{aligned} \tag{C.7}$$

where $M_n = M_n(T) = M(\tau_n + T)$ is defined as in (C.2). Now, mimicking (C.6), let

$$\Delta_0(t;T) = \sup_{0\le h\le T}\left\|\int_t^{t+h}\overline{U}(s)\,ds\right\|. \tag{C.8}$$

so our previous bound shows that

$$\mathbb{E}[\Delta_0(t;T)^2] \le C_2\int_t^{t+T}\overline{\gamma}^2(s)\overline{\sigma}^2(s)\,ds. \tag{C.9}$$

We will proceed to show that $\lim_{t\to\infty}\Delta_0(t;T) = 0$ for all $T > 0$ by considering the sequence of intervals $[kT, (k+1)T]$ and using the Borel-Cantelli lemma to show that $\Delta_0(kT;T) \to 0$ as $k \to \infty$. Indeed, we have

$$\sum_{k=1}^{\infty}\mathbb{E}[\Delta_0(kT;T)^2] \le C_2\int_0^{\infty}\overline{\gamma}^2(s)\overline{\sigma}^2(s)\,ds = C_2\sum_{n=1}^{\infty}\gamma_n^2\sigma_n^2 < \infty \tag{C.10}$$

with the last step following from Assumption (A5.2). Then, if we consider the event $\mathcal{E}_k(\varepsilon) = \{\Delta_0(kT;T) > \varepsilon\}$, Chebysev's inequality gives

$$\sum_{k=1}^{\infty}\mathbb{P}(\mathcal{E}_k(\varepsilon)) \le \frac{\sum_{k=1}^{\infty}\mathbb{E}[\Delta_0(kT;T)^2]}{\varepsilon^2} < \infty, \tag{C.11}$$

and hence, by the Borel-Cantelli lemma, we get

$$\mathbb{P}\left(\limsup_{k\to\infty}\mathcal{E}_k(\varepsilon)\right) = 0. \tag{C.12}$$

This shows that, with probability 1, we have $\Delta_0(kT;T) \le \varepsilon$ for all but a finite number of k; put differently, the event $\mathcal{E}(\varepsilon) = \{\Delta_0(kT;T) \text{ occurs infinitely often}\} = \bigcap_{n=1}^{\infty}\bigcup_{k=n}^{\infty}\mathcal{E}_k(\varepsilon)$ has $\mathbb{P}(\mathcal{E}(\varepsilon)) = 0$. Therefore, as a union of probability zero events, we have

$$\mathbb{P}\left(\liminf_{k\to\infty}\Delta_0(kT;T) > 0\right) = \mathbb{P}\left(\bigcup_{n=1}^{\infty}\mathcal{E}(1/n)\right) \le \sum_{n=1}^{\infty}\mathbb{P}(\mathcal{E}(1/n)) = 0, \tag{C.13}$$

i.e., $\Delta_0(kT;T) \to 0$ with probability 1.

Thus, going back to the requirements of Proposition C.1, we get

$$\begin{aligned}\Delta(kT;T) &= \sup_{0\le h\le T}\left\|\int_{kT}^{kT+h}\overline{W}(t)\,dt\right\| = \sup_{0\le h\le T}\left\|\int_{kT}^{kT+h}[\overline{U}(t)+\overline{b}(t)]\,dt\right\| \\ &\le \Delta_0(kT;T) + \sup_{0\le h\le T}\int_{kT}^{kT+h}\overline{B}(t)\,dt. \\ &\le \Delta_0(kT;T) + T\max_{0\le h\le T}\overline{B}(kT+h). \end{aligned}\tag{C.14}$$

Given that $\lim_{k\to\infty} B_k = 0$, the above shows that $\Delta(kT;T)\to 0$ as $k\to\infty$. Moreover, for all $t\in[kT,(k+1)T]$, we have $\Delta(t;T)\le 2\Delta(kT;T)+\Delta((k+1)T;T)$ so $\Delta(t;T)\to 0$ with probability 1. With $T>0$ arbitrary, we conclude that (C.5) holds with probability 1, and our claim follows from Proposition C.1. ■

To proceed, it will be convenient to consider a stronger version of Assumption (A5.2):

$$\mathbb{P}(\|U_n\|\ge t\,|\,\mathcal{F}_n)\le 2e^{-\frac{t^2}{2\sigma^2}} \tag{A5.2$'$}$$

for some $\sigma\ge 0$ and all $n=1,2,\dots$, $t\ge 0$. Some of the RM schemes presented in Section 4.3 will verify this stronger criterion; see Appendix C.1.3 below.

Under this assumption, we obtain the following generalization of a criterion due to [BH96]:

Proposition C.2. *Suppose that* (RM) *is run with a step-size policy* γ_n *such that* $A/n\le\gamma_n\le B/(\log n)^{1+\varepsilon}$ *for some* $B,\varepsilon>0$. *If Assumptions* (A5.1) and (A5.2$'$) *hold, then, with probability* 1, *a*) Z_n *is an APT of* (MD)*; or b*) Z_n *is unbounded* (*and hence, non-convergent*).

Proof. As in the proof of Theorem 4.1, our approach will hinge on the proviso (C.5) of Proposition C.1 and, in particular, controlling the quantity $\Delta_0(t;T)$ defined in (C.8). We proceed step-by-step:

Step 1: A union bound for the tails of $\sup_{n\le m\le M_n}\|\sum_{k=n}^m\gamma_k U_k\|$**.** Up to a multiplicative constant that depends only on the dimension of the problem, we can assume without loss of generality that $\|\cdot\|$ is the sup-norm $\|z\|=\max_i|z_i|$. In this case, we have $\|z\|\ge t$ if and only if there exists a basis vector e_i of $\mathbb{R}^d$ such that $\langle z,e_i\rangle\ge t$ or $\langle z,e_i\rangle\le -t$. We thus get the union bound

$$\begin{aligned}\mathbb{P}\left(\sup_{n\le m\le M_n}\left\|\sum_{k=n}^m\gamma_k U_k\right\|\ge t\right) &\le \sum_{i=1}^d\mathbb{P}\left(\sup_{n\le m\le M_n}\sum_{k=n}^m\langle\gamma_k U_k,e_i\rangle\ge t\right) \\ &\quad+\sum_{i=1}^d\mathbb{P}\left(\sup_{n\le m\le M_n}\sum_{k=n}^m\langle\gamma_k U_k,-e_i\rangle\ge t\right).\end{aligned}\tag{C.15}$$

In view of this, we will focus below on the tail probability $\mathbb{P}(\sup_{n\le m\le M_n}\sum_{k=n}^m\langle\gamma_k U_k,z\rangle)$ for arbitrary $z\in\mathbb{R}^d$.

Step 2: Exponential tail concentration. By standard arguments, Assumption (A5.2′) is equivalent to asking that

$$\mathbb{E}[\exp(\langle z, U_n\rangle) \,|\, \mathcal{F}_n] \leq \exp(\sigma^2 \|z\|^2/2). \tag{C.16}$$

With this reformulation in mind, consider the process

$$Q_n(z) = \exp\left(\sum_{k=1}^{n} \langle z, \gamma_k U_k\rangle - \frac{\sigma^2}{2} \sum_{k=1}^{n} \gamma_k^2 \|z\|^2\right). \tag{C.17}$$

Then, by construction

$$\begin{aligned}\mathbb{E}[Q_n(z) \,|\, \mathcal{F}_n] &= \mathbb{E}\left[\exp\left(\sum_{k=1}^{n} \langle z, \gamma_k U_k\rangle - \frac{\sigma^2}{2} \sum_{k=1}^{n} \gamma_k^2 \|z\|^2\right) \middle|\, \mathcal{F}_n\right] \\ &= Q_{n-1}(z)\,\mathbb{E}\left[\exp\left(\langle z, \gamma_n U_n\rangle - \frac{\sigma^2}{2}\gamma_n^2 \|z\|^2\right) \middle|\, \mathcal{F}_n\right] \leq Q_{n-1}(z),\end{aligned} \tag{C.18}$$

i.e., $Q_n(z)$ is a supermartingale relative to $\mathcal{F}_n$.[2] Moreover, we have:

$$\begin{aligned}\mathbb{P}\left(\sup_{n\leq m\leq M_n} \sum_{k=n}^{m} \langle \gamma_k U_k, z\rangle \geq \alpha\right) &= \mathbb{P}\left(\sup_{n\leq m\leq M_n} \frac{Q_m(z)}{Q_n(z)} \exp\left(\frac{\sigma^2}{2} \sum_{k=n}^{m} \gamma_k^2 \|z\|^2\right) \geq \exp(\alpha)\right) \\ &= \mathbb{P}\left(\sup_{n\leq m\leq M_n} \frac{Q_m(z)}{Q_n(z)} \exp\left(\frac{\sigma^2}{2} \sum_{k=n}^{M_n} \gamma_k^2 \|z\|^2\right) \geq \exp(\alpha)\right) \\ &= \mathbb{P}\left(\sup_{n\leq m\leq M_n} \frac{Q_m(z)}{Q_n(z)} \geq \exp\left(\alpha - \frac{\sigma^2}{2} \sum_{k=n}^{M_n} \gamma_k^2 \|z\|^2\right)\right) \\ &\leq \mathbb{E}\left[\sup_{n\leq m\leq M_n} \frac{Q_m(z)}{Q_n(z)}\right] \cdot \exp\left(\frac{\sigma^2}{2} \sum_{k=n}^{M_n} \gamma_k^2 \|z\|^2 - \alpha\right) \\ &\leq \exp\left(\frac{\sigma^2}{2} \sum_{k=n}^{M_n} \gamma_k^2 \|z\|^2 - \alpha\right)\end{aligned} \tag{C.19}$$

where we used Markov's inequality in the last step and the fact that $Q_n(z)$ is a submartingale in the penultimate one. Thus, letting $\Sigma = \sigma^2 \sum_{k=n}^{M_n} \gamma_k^2 \|z\|^2$ and taking $z \leftarrow (t/\Sigma)e_i$, $t \leftarrow t^2/\Sigma$, we get

$$\mathbb{P}\Big(\sup_{n\leq m\leq M_n} \sum_{k=n}^{m} \langle \gamma_k U_k, e_i\rangle \geq t\Big) \leq \exp\left(-\frac{\sigma^2 t^2}{2\sum_{k=n}^{M_n} \gamma_k^2}\right). \tag{C.20}$$

---

[2] Recall here that, by the definition of the filtration $\mathcal{F}_n$, U_n is $\mathcal{F}_{n+1}$-measurable but not $\mathcal{F}_n$-measurable.

Step 3: Closing the gap. By assumption, $\sum_{k=n}^{M_n} \gamma_n^2 \leq T\gamma_n^2 \leq T/(\log n)^{2+2\varepsilon}$. Hence

$$\exp\left(-\frac{\sigma^2 t^2}{2\sum_{k=n}^{M_n} \gamma_k^2}\right) \leq \exp\left(-\frac{\sigma^2}{2}\frac{(\log n)^{2+2\varepsilon}}{T}\right) = n^{-\frac{\sigma^2}{2}\frac{(\log n)^{1+2\varepsilon}}{T}}. \tag{C.21}$$

Therefore

$$\mathbb{P}\left(\sup_{n\leq m\leq M_n}\left\|\sum_{k=n}^{m} \gamma_k U_k\right\| \geq t\right) \leq \frac{C_2'}{n^2} \tag{C.22}$$

for some suitable constant $C_2' > 0$. With notation as in the proof of Theorem 4.1, this implies that

$$\sum_{k=1}^{\infty} \mathbb{P}(\Delta_0(kT;T) \leq \alpha) = \mathcal{O}\left(\sum_{k=1}^{\infty}\frac{1}{k^2}\right) < \infty. \tag{C.23}$$

Thus, by applying the Borel-Cantelli lemma as in the proof of Theorem 4.1, we conclude that $\Delta_0(kT;T) \to 0$ with probability 1. The rest of the arguments required to show that $\lim_{t\to 0}\Delta(t;T) = 0$ for all T follow the lines of the proof of Theorem 4.1, so we omit them. ■

C.1.3 Proof of Proposition 4.1

We are now in a position to prove that the generalized RM schemes presented in Section 4.3 comprise asymptotic pseudotrajectories of the mean dynamics (MD). For convenience, we state the relevant result below:

Proposition 4.1. *Let Z_n be a sequence generated by any of the Algorithms 4.1–4.5. Assume further that:*

a) *For first-order methods* (*Algorithms 4.1–4.4*), *the algorithm is run with SFO feedback satisfying* (4.2) *and a step-size γ_n such that $A/n \leq \gamma_n \leq B/(\log n)^{1+\varepsilon}$ for some $A, B, \varepsilon > 0$.*

b) *For zeroth-order methods* (*Algorithm 4.5*), *the algorithm is run with parameters γ_n and δ_n such that $\lim_n(\gamma_n + \delta_n) = 0$, $\sum_n \gamma_n = \infty$, and $\sum_n \gamma_n^2/\delta_n^2 < \infty$ (e.g., $\gamma_n = 1/n$, $\delta_n = 1/n^{1/3}$).*

Then, with probability 1, *one of the following holds: a*) *Z_n is an APT of* (MD)*; or b*) *Z_n is unbounded.*

Proof. We proceed method-by-method:

Algorithm 4.1: Stochastic gradient descent/ascent. For (SGDA), we have $W_n = U_n = \mathsf{U}(\omega_n)$ and $b_n = 0$, so Assumption (A5.1) is satisfied automatically (since $B_n = 0$). Moreover, under the stated assumptions for (SFO), U_n is sub-Gaussian, so our claim follows from Proposition C.2.

Algorithm 4.2: Alternating stochastic gradient descent/ascent. For (alt-SGDA), we have $b_n = (0, V_{\boldsymbol{y}}(X_{n+1}, Y_n) - V_{\boldsymbol{y}}(X_n, Y_n))$, and $U_n = (U_{\boldsymbol{x},n}, U_{\boldsymbol{y},n})$. Under the stated assumptions for (SFO), U_n satisfies Assumption (A5.2′), so we are left to show that Assumption (A5.1) holds, i.e., that $b_n \to 0$. To that end, since V is Lipschitz, we have

$$\|b_n\| = \|V_{\boldsymbol{y}}(X_{n+1}, Y_n) - V_{\boldsymbol{y}}(X_n, Y_n)\| \leq L\|X_{n+1} - X_n\|, \tag{C.24}$$

where L denotes the Lipschitz modulus of V. Hence, by the definition of (alt-SGDA), we get

$$\|b_n\| \leq \gamma_n L\|V_{\boldsymbol{y}}(X_{n+1}, Y_n) + U_{\boldsymbol{y},n}\| \leq \gamma_n L\|V_{\boldsymbol{y}}(X_{n+1}, Y_n)\| + \gamma_n L\|U_{\boldsymbol{y},n}\| \tag{C.25}$$

If Z_n is bounded, we also have $\sup_n \|V_{\boldsymbol{y}}(X_{n+1}, Y_n)\| < \infty$, so the first term above vanishes as $n \to \infty$ (recall that $\lim_n \gamma_n = 0$). As for the second, we have

$$\mathbb{P}(\|U_n\| \geq \log n) \leq 2e^{-(\log n)^2/(2\sigma^2)} = 2n^{-\log n/(2\sigma^2)} \tag{C.26}$$

In turn, this implies that $\sum_{n=1}^{\infty} \mathbb{P}(\|U_n\| \geq \log n) < \infty$ so, by the Borel-Cantelli lemma, we have $\|U_n\| = \mathcal{O}(\log n)$ with probability 1. Hence, by our assumptions for the method's step-size, we get

$$\gamma_n \|U_{\boldsymbol{y},n}\| \leq \gamma_n \|U_n\| = \mathcal{O}\left(\frac{\log n}{(\log n)^{1+\varepsilon}}\right) = \mathcal{O}\left(\frac{1}{(\log n)^{\varepsilon}}\right) \tag{C.27}$$

i.e., $B_n \to 0$ with probability 1. Our claim then follows from Proposition C.2.

Algorithm 4.3: Stochastic extra-gradient. For (SEG), we have $U_n = \mathsf{U}(Z_n^+; \omega_n^+)$ and $b_n = V(Z_n^+) - V(Z_n)$, so Assumption (A5.2′) holds by default. For Assumption (A5.1), arguing as in the case of Algorithm 4.2 above, we have

$$\begin{aligned}\|b_n\| = \|V(Z_n^+) - V(Z_n)\| \leq L\|Z_n^+ - Z_n\| \\ &= \gamma_n \|\mathsf{V}(\omega_n)\| = \gamma_n L\|V(Z_n) + \mathsf{U}(\omega_n)\| \\ &\leq \gamma_n L\|V(Z_n)\| + \gamma_n L\|\mathsf{U}(\omega_n)\|,\end{aligned} \tag{C.28}$$

Thus, by Proposition C.2, we conclude that Z_n is an APT of (MD).

Algorithm 4.4: Optimistic gradient. For (OG/PEG), we have $U_n = \mathsf{U}(\omega_n^+)$ and $b_n = V(Z_n^+) - V(Z_n)$. so Assumption (A5.2′) again holds by default. The bias term can then be bounded exactly as in the case of Algorithm 4.3, so our APT claim follows again by Proposition C.2.

Algorithm 4.5: Simultaneous perturbation stochastic approximation. Because of the algorithm's different oracle structure (zeroth- vs. first-order feedback), the analysis of (SPSA) is

different. We begin with the algorithm's bias term, given here by

$$b_n = \mathbb{E}[V_n \,|\, \mathcal{F}_n] - V(Z_n) \tag{C.29}$$

with

$$V_n = \pm(d/\delta_n)\, F(Z_n + \delta_n \omega_n)\, \omega_n \tag{C.30}$$

denoting the method's one-shot SPSA estimator. To bound it, let

$$v_{i,n} = \mathbb{E}[V_{i,n} \,|\, \mathcal{F}_n] \tag{C.31}$$

denote the i-th component of $V_n \in \mathbb{R}^d$ after having averaged out the choice of the random seed ω_n (which, by default, is not $\mathcal{F}_n$-measurable). We then have

$$v_{i,n} = \pm \frac{d}{\delta_n} \cdot \frac{1}{2d} \big[F(Z_n + \delta_n e_i) - F(Z_n - \delta_n e_i) \big] \tag{C.32}$$

where, as per our discussion in Section 4.3, the "±" sign is equal to -1 if $e_i \in \mathcal{E}_{\mathcal{X}}$ and $+1$ if $e_i \in \mathcal{E}_{\mathcal{Y}}$. Then, by the mean value theorem, there exists some $\tilde{Z}_n$ in the line segment $[Z_n - \delta_n e_i, Z_n + \delta_n e_i]$ such that

$$v_{i,n} = \pm \partial_i F(\tilde{Z}_n) = V_{i,n}(\tilde{Z}_n). \tag{C.33}$$

Since V is Lipschitz continuous, it follows that

$$|v_{i,n} - V_{i,n}(Z_n)| = |V_{i,n}(\tilde{Z}_n) - V_{i,n}(Z_n)| \le L \|\tilde{Z}_n - Z_n\| = \mathcal{O}(\delta_n) \tag{C.34}$$

since $\tilde{Z}_n \in [Z_n - \delta_n e_i, Z_n + \delta_n e_i]$. Finally, for the oracle's variance, we have $\|V_n\|^2 = \mathcal{O}(1/\delta_n^2)$ by construction so, under the stated assumptions for γ_n and δ_n, Assumption (A5.2) is satisfied and our claim follows from Theorem 4.1. ■

We conclude this appendix with a simple coercivity criterion which guarantees that the iterates of an iterative method of the general form (RM) remain bounded:

Proposition C.3. *Suppose that V satisfies the coercivity condition*

$$\liminf_{\|z\|\to\infty} \frac{\langle V(z), z\rangle}{\|z\|^2} < 0. \tag{A5.3}$$

Then, under Assumptions (A5.1) and (A5.2)*, the sequence Z_n generated by* (RM) *is bounded* (*a.s.*).

Corollary C.1. *Under Assumptions* (A5.1)–(A5.3)*, the iterates Z_n of* (RM) *comprise an APT of* (MD).

Proof. To begin, observe that, under Assumption (A5.3), the quadratic penalty function $E(z) =$

$\sum_i z_i^2/2$ is a Lyapunov function for (MD) as $\|z\| \to \infty$. Indeed, by Assumption (A5.3), there exists some $R > 0$ such that, whenever $\|z\| \geq R$, we have

$$\frac{dE}{dt} = \langle \nabla E(z), \dot{z} \rangle = \langle \nabla E(z), V(z) \rangle \leq -\frac{\kappa}{2} \|z\|^2 \tag{C.35}$$

where $\kappa = -\liminf_{\|z\|\to\infty} \langle V(z), z \rangle / \|z\|^2 > 0$.[3] This shows that trajectories of (MD) cannot escape to infinity so it is plausible to expect the same to hold for (RM).

Our proof of this fact follows a direct stabilization technique due to [KY97]. Specifically, going back to (RM), a simple expansion gives

$$\begin{aligned} E(Z_{n+1}) &= E(Z_n) + \gamma_n \langle V_n, Z_n \rangle + \frac{1}{2} \gamma_n^2 \|V_n\|^2 \\ &\leq E(Z_n) + \gamma_n \langle V(Z_n), Z_n \rangle + \gamma_n \langle W_n, Z_n \rangle + \gamma_n^2 \|V_n\|^2 \end{aligned} \tag{C.36}$$

Hence, taking (conditional) expectations, we obtain:

$$\mathbb{E}[E(Z_{n+1}) \,|\, \mathcal{F}_n] \leq E(Z_n) + \gamma_n \langle V(Z_n) + b_n, Z_n \rangle + \gamma_n^2 \mathbb{E}[\|V_n\|^2 \,|\, \mathcal{F}_n]. \tag{C.37}$$

To proceed, note that, by Assumptions (A5.1) and (A5.2), we have

$$\mathbb{E}\left[\sum_{n=1}^{\infty} \gamma_n^2 \|V_n^2\| \mathbb{1}_{\{\|Z_n\| \leq R\}} \right] < \infty, \tag{C.38}$$

while, otherwise

$$\mathbb{E}\big[\|V_n\|^2 \,\big|\, \mathcal{F}_n\big] \leq C\big(\sigma_n^2 + (\kappa/2)\|Z_n\|^2\big) \quad \text{whenever } \|Z_n\| \geq R. \tag{C.39}$$

Consider now the process

$$S_n = \mathbb{E}\big[\textstyle\sum_{k\geq n} \gamma_k^2 \|V_n\|^2 \mathbb{1}_{\{\|Z_k\| \leq R\}} \,\big|\, \mathcal{F}_n\big] \tag{C.40}$$

and let $E_n = E(Z_n) + S_n$. By definition, E_n is non-negative; moreover, by (C.36), we get

$$\mathbb{E}[E_{n+1} - E_n \,|\, \mathcal{F}_n] \leq -\frac{\kappa \gamma_n}{2} \|Z_n\|^2 + \frac{C\gamma_n^2}{2} \|Z_n\|^2. \tag{C.41}$$

Since $\gamma_n \to 0$, it follows that E_n is eventually a supermartingale: specifically, if $n_0 = \sup\{n : C\gamma_n > \kappa\}$ (with the standard convention $\sup \varnothing = -\infty$), we have $\mathbb{E}[E_{n+1} \,|\, \mathcal{F}_n] \leq E_n$ for all $n \geq n_0$. Since $\mathbb{E}[E_{n_0}] < \infty$, Doob's submartingale convergence theorem subsequently implies that E_n converges with probability 1 to some non-negative random variable E_∞. Since $S_n \to 0$ with probability 1 (by Assumption (A5.2)), we conclude that $\|Z_n\| = (2/\kappa)E(Z_n) \to (2/\kappa)E_\infty$ (a.s.), and our claim follows. ■

---

[3] In the above and throughout this proof, we assume that $\|\cdot\|$ is the ordinary Euclidean norm on $\mathbb{R}^d$; this assumption is only made for notational convenience and to avoid carrying around many multiplicative constants.

C.2 Convergence analysis

With all this preliminary work in hand, we are finally in a position to prove Theorems 4.2 and 4.3. The heavy lifting for the former is provided by the fact that, under the requirements of Theorem 4.1 and/or Proposition 4.1, Z_n is an APT of the mean dynamics (MD), so it inherits its limit structure. The latter requires completely different techniques and involves a much finer analysis of the process in hand.

C.2.1 Convergence to ICTs

We begin with Theorem 4.2, which we restate below for convenience:

Theorem 4.2. *Suppose that* (RM) *is run with a step-size sequence* γ_n *such that* $\sum_n \gamma_n = \infty$, $\lim_n \gamma_n = 0$. *If Assumptions* (A5.1) and (A5.2) *hold, then, with probability* 1*, we have: a*) Z_n *converges to an ICT set of* F*; or b*) Z_n *is unbounded* (*and hence, non-convergent*).

Proof. We consider two cases. First, if Z_n is unbounded, there is nothing to show. Otherwise, if Z_n is bounded, Theorem 4.2 shows that it is an APT of the mean dynamics (MD). Now, let $\mathcal{L} = \bigcap_{t\geq 0} \operatorname{cl}(Z(t,\infty))$ be the limit set of $Z(t)$, i.e., the set of limit points of convergent sequences $Z(t_n)$ with $\lim_n t_n = \infty$. Our claim then follows by the limit set theorem of [BH96, Theorem 8.2]. ■

As we discussed in the main part of our paper, the ICT sets of F may exhibit a wide variety of structural properties (limit cycles, heteroclinic networks, etc.). As a complement to this, we show below that, in *gradient* systems ($V = -\nabla f$ for some $f\colon \mathcal{Z} \to \mathbb{R}$), ICT sets can only be compoments of equilibria. Specifically, building on a general result by [Ben99], we have:

Proposition C.4. *Suppose that* $V(z) = -\nabla f(z)$ *for some* C^d*-smooth potential function* $f\colon \mathcal{Z} \to \mathbb{R}$ *with a compact critical set* $\operatorname{crit}(f) = \{z^\star : \nabla f(z^\star) = 0\}$*. Then, every ICT set* $\mathcal{S}$ *of* (MD) *is contained in* $\operatorname{crit}(f)$*; moreover,* f *is constant on* $\mathcal{S}$*. In particular, any ICT set of* (MD) *consists solely of critical points of* f*.*

Proof. Under the stated conditions, the critical set $\mathcal{Z}^\star := \operatorname{crit}(f)$ of f coincides with the set of rest points of (MD). Moreover, by Sard's theorem [Lee03], $f(\mathcal{Z}^\star)$ has zero Lebesgue measure and hence empty interior. Our claim then follows from Proposition 6.4 of [Ben99]. ■

As another elementary illustration in addition to the gradient systems, one can show that for bilinear games $F(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}\boldsymbol{y}$, the ICT sets are annular regions of the form $\{z : r \leq \|z\| \leq R,\ 0 \leq r \leq R\}$. This can be easily seen by considering the widely known Hamiltonian function $H(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}^2 + \boldsymbol{y}^2$, which satisfies $\dot{H} = 0$ provided $(\boldsymbol{x},\boldsymbol{y})$ follows (MD). An immediate consequence of this fact is that *any* point on $\mathbb{R}^2$ lies in some ICT set of (MD), which further implies that there is no bounded attracting region, i.e., attractors.

C.2.2 Convergence to attractors

We now proceed with the analysis of RM schemes in the presence of an attractor; the relevant result is Theorem 4.3:

Theorem 4.3. *Let $\mathcal{S}$ be an attractor of* (MD) *and fix some confidence level $\alpha > 0$. If γ_n is small enough and Assumptions* (A5.1) and (A5.2) *hold, there exists a neighborhood $\mathcal{U}$ of $\mathcal{S}$, independent of α, such that $\mathbb{P}(Z_n$ converges to $\mathcal{S} \mid Z_1 \in \mathcal{U}) \geq 1 - \alpha$.*

Because of the generality of our assumptions, the proof of Theorem 4.3 requires a range of completely different arguments and techniques. We illustrate the main steps of our technical trajectory below:

1. The first crucial component of our proof is to establish an energy function for (RM) in a neighborhood of $\mathcal{S}$. To do this, we rely on Conley's decomposition theorem (the so-called "fundamental theorem of dynamical systems") which states that the mean dynamics (MD) are "gradient-like" in a neighborhood of an attractor, i.e., they admit a (local) Lyapunov function.
2. Because of the noise in (RM), the evolution of E along the trajectories of (RM) could present *signifcant* jumps: in particular, a single "bad" realization of the noise could carry Z_n out of the basin of attraction of $\mathcal{S}$, possibly never to return. A major difficulty here is that the driving vector field V is *not* assumed bounded, so it is not straightforward to establish proper control over the error terms of (RM). However, we show that, with high probability (and, in particular, with probability at least $1 - \alpha$), the aggregation of these errors remains controllably small; this is the most technically challenging part of our argument and it unfolds in a series of lemmas below.
3. Conditioning on the above, we will show that, with probability at least $1 - \alpha$, the value of the trajectory's energy cannot grow more than a token threshold ε; as a result, if (RM) is initialized close to $\mathcal{S}$, it will remain in a neighborhood thereof for all n (again, with probability at least $1 - \alpha$).
4. Thanks to this "stochastic Lyapunov stability" result, we can regain control of the variance of the process and use martingale limit and maximal inequality arguments to show that Z_n converges to $\mathcal{S}$.

In the rest of this section, we make this roadmap precise via a series of technical lemmas and intermediate results.

A local energy function for (RM). We begin by providing a suitable (local) energy function for (MD). Indeed, since $\mathcal{S}$ is an attractor, there exists a compact neighborhood $\mathcal{K}$ of $\mathcal{S}$, called the *fundamental neighborhood* of $\mathcal{S}$, and having the defining property that $\operatorname{dist}(\Theta_t(z), \mathcal{S}) \to 0$ as $t \to \infty$ uniformly in $z \in \mathcal{K}$. Since all trajectories of (MD) that start in $\mathcal{K}$ converge to $\mathcal{S}$, there are no other non-trivial invariant sets in $\mathcal{K}$ except $\mathcal{S}$. As a result, with $\mathcal{K}$ compact, Conley's decomposition theorem for dynamical systems [Con78] shows that there exists a smooth Lyapunov – or "energy" – function $E\colon \mathcal{K} \to \mathbb{R}$ such that (*i*) $E(z) \geq 0$ with equality if and only if $z \in \mathcal{S}$; and (*ii*) $\dot{E}(z) := \langle \nabla E(z), V(z)\rangle < 0$ for all $z \in \mathcal{K} \setminus \mathcal{S}$ (implying in particular that $E(\Theta_t(z))$ is strictly decreasing in t whenever $z \in \mathcal{K} \setminus \mathcal{S}$).

In the discrete-time context of (RM), the energy $E_n := E(Z_n)$ of Z_n may fail to be decreasing (strictly or otherwise). However, a simple Taylor expansion with Lagrange remainder yields the basic energy bound

$$E_{n+1} \leq E_n + \gamma_n \langle \nabla E(Z_n), V(Z_n)\rangle + \gamma_n \xi_n + \gamma_n \psi_n + \gamma_n^2 \theta_n^2, \tag{C.42}$$

where the error terms ξ_n, ψ_n and θ_n are defined as

$$\xi_n = \langle \nabla E(Z_n), U_n\rangle \tag{C.43a}$$

$$\psi_n = B_n \|\nabla E(Z_n)\| + \gamma_n \beta B_n^2 \tag{C.43b}$$

$$\theta_n^2 = \beta \|V(Z_n) + U_n\|^2 \tag{C.43c}$$

with β denoting the strong smoothness modulus of E over the compact set $\mathcal{K}$. Clearly, each of these error terms can be positive, so E_n may fail to be decreasing; we discuss how these errors can be controlled below.

Error control. We begin by encoding the aggregation of the error terms in (C.42) as

$$M_n = \sum_{k=1}^{n} \gamma_k \xi_k \tag{C.44a}$$

and

$$S_n = \sum_{k=1}^{n} [\gamma_k \psi_k + \gamma_k^2 \theta_k^2] \tag{C.44b}$$

Since $\mathbb{E}[\xi_n \,|\, \mathcal{F}_n] = 0$, we have $\mathbb{E}[M_n \,|\, \mathcal{F}_n] = M_{n-1}$, so M_n is a martingale; likewise, $\mathbb{E}[S_n \,|\, \mathcal{F}_n] \geq S_{n-1}$, so S_n is a submartingale. Interestingly, even though M_n appears more "balanced" as an error (because ξ_n is zero-mean), it is more difficult to control because the variance of its increments is

$$\mathbb{E}[|\gamma_n \xi_n|^2 \,|\, \mathcal{F}_n] = \gamma_n^2 \mathbb{E}[|\langle \nabla E(Z_n), U_n\rangle|^2 \,|\, \mathcal{F}_n], \tag{C.45}$$

so the jumps of M_n can become arbitrarily big if Z_n escapes $\mathcal{K}$ (which is the event we are trying to discount in the first place). On that account, we will instead bound the total error increments by *conditioning* everything on the event that Z_n remains within $\mathcal{K}$.

To make this precise, consider the "mean square" error process

$$R_n = M_n^2 + S_n \tag{C.46}$$

and the indicator events

$$\mathcal{E}_n \equiv \mathcal{E}_n(\mathcal{K}) = \{Z_n \in \mathcal{K} \text{ for all } k = 1,2,\dots,n\} \tag{C.47}$$

$$\mathcal{H}_n \equiv \mathcal{H}_n(\varepsilon) = \{R_k \leq \varepsilon \text{ for all } k = 1,2,\dots,n\}, \tag{C.48}$$

with the convention $\mathcal{E}_0 = \mathcal{H}_0 = \Omega$. Moving forward, with significant hindsight, we will choose ε small enough so that

$$\{z \in \mathcal{Z} : E(z) \leq 2\varepsilon + \sqrt{\varepsilon}\} \subseteq \mathcal{K}. \tag{C.49}$$

and we will assume that Z_1 is initialized in a neighborhood $\mathcal{U} \subseteq \mathcal{K}$ such that

$$\mathcal{U} \subseteq \{z \in \mathcal{Z} : E(z) \leq \varepsilon\} \tag{C.50}$$

We then have the following estimates:

Lemma C.1. *Suppose that $Z_1 \in \mathcal{U}$ and Assumptions* (A5.1) and (A5.2) *hold. Then*

1. $\mathcal{E}_{n+1} \subseteq \mathcal{E}_n$ *and* $\mathcal{H}_{n+1} \subseteq \mathcal{H}_n$.
2. $\mathcal{H}_{n-1} \subseteq \mathcal{E}_n$.
3. *Consider the "bad realization" event*

$$\begin{aligned}\tilde{\mathcal{H}}_n := \mathcal{H}_{n-1} \setminus \mathcal{H}_n = \mathcal{H}_{n-1} \cap \{R_n > \varepsilon\} \\ = \big\{R_k \leq \varepsilon \textit{ for } k = 1,2,\dots,n-1 \textit{ and } R_n > \varepsilon\big\},\end{aligned} \tag{C.51}$$

and let $\tilde{R}_n = R_n \mathbb{1}_{\mathcal{H}_{n-1}}$ denote the cumulative error subject to the noise being "small" until time n. Then:

$$\mathbb{E}[\tilde{R}_n] \leq \mathbb{E}[\tilde{R}_{n-1}] + \gamma_n G B_n + \gamma_n^2[2\beta G^2 + (2\beta + G^2)\sigma_n^2 + \beta B_n^2] - \varepsilon\,\mathbb{P}(\tilde{\mathcal{H}}_{n-1}), \tag{C.52}$$

where $G^2 = \sup_{z\in\mathcal{K}}\{\|\nabla E(z)\|^2 + \|V(z)\|^2\}$ and, by convention, $\tilde{\mathcal{H}}_0 = \varnothing$, $\tilde{R}_0 = 0$.

Proof. The first claim is obvious. For the second, we proceed inductively:

1. For the base case $n = 1$, we have $\mathcal{E}_1 = \{Z_1 \in \mathcal{K}\} \supseteq \{Z_1 \in \mathcal{U}\} = \Omega$ (recall that Z_1 is initialized in $\mathcal{U} \subseteq \mathcal{K}$). Since $\mathcal{H}_0 = \Omega$, our claim follows.
2. Inductively, suppose that $\mathcal{H}_{n-1} \subseteq \mathcal{E}_n$ for some $n \geq 1$. To show that $\mathcal{H}_n \subseteq \mathcal{E}_{n+1}$, suppose that $R_k \leq \varepsilon$ for all $k = 1,2,\dots,n$. Since $\mathcal{H}_n \subseteq \mathcal{H}_{n-1}$, this implies that $\mathcal{E}_n$ also occurs, i.e., $Z_k \in \mathcal{K}$ for all $k = 1,2,\dots,n$; as such, it suffices to show that $Z_{n+1} \in \mathcal{K}$.

To do so, given that $Z_k \in \mathcal{U} \subseteq \mathcal{K}$ for all $k = 1, 2, \dots n$, the bound (C.42) gives

$$E_{k+1} \le E_k + \gamma_n \xi_n + \gamma_n \psi_n + \gamma_n^2 \theta_n^2, \quad \text{for all } k = 1, 2, \dots n, \tag{C.53}$$

and hence, after telescoping over $k = 1, 2, \dots, n$, we get

$$E_{n+1} \le E_1 + M_n + S_n \le E_1 + \sqrt{R_n} + R_n \le \varepsilon + \sqrt{\varepsilon} + \varepsilon = 2\varepsilon + \sqrt{\varepsilon}. \tag{C.54}$$

We conclude that $E(Z_{n+1}) \le 2\varepsilon + \sqrt{\varepsilon}$, i.e., $Z_{n+1} \in \mathcal{K}$, as required for the induction.

For our third claim, note first that

$$\begin{aligned} R_n &= (M_{n-1} + \gamma_n \xi_n)^2 + S_{n-1} + \gamma_n \psi_n + \gamma_n^2 \theta_n^2 \\ &= R_{n-1} + 2\gamma_n \xi_n M_{n-1} + \gamma_n^2 \xi_n^2 + \gamma_n \psi_n + \gamma_n^2 \theta_n^2, \end{aligned} \tag{C.55}$$

so, after taking expectations:

$$\mathbb{E}[R_n \,|\, \mathcal{F}_n] = R_{n-1} + 2M_{n-1}\gamma_n \mathbb{E}[\xi_n \,|\, \mathcal{F}_n] + \mathbb{E}[\gamma_n^2 \xi_n^2 + \gamma_n \psi_n + \gamma_n^2 \theta_n^2 \,|\, \mathcal{F}_n] \ge R_{n-1} \tag{C.56}$$

i.e., R_n is a submartingale. To proceed, let $\tilde{R}_n = R_n \mathbb{1}_{\mathcal{H}_{n-1}}$ so

$$\begin{aligned} \tilde{R}_n &= R_{n-1} \mathbb{1}_{\mathcal{H}_{n-1}} + (R_n - R_{n-1}) \mathbb{1}_{\mathcal{H}_{n-1}} \\ &= R_{n-1} \mathbb{1}_{\mathcal{H}_{n-2}} - R_{n-1} \mathbb{1}_{\tilde{\mathcal{H}}_{n-1}} + (R_n - R_{n-1}) \mathbb{1}_{\mathcal{H}_{n-1}}, \\ &= \tilde{R}_{n-1} + (R_n - R_{n-1}) \mathbb{1}_{\mathcal{H}_{n-1}} - R_{n-1} \mathbb{1}_{\tilde{\mathcal{H}}_{n-1}}, \end{aligned} \tag{C.57}$$

where we used the fact that $\mathcal{H}_{n-1} = \mathcal{H}_{n-2} \setminus \tilde{\mathcal{H}}_{n-1}$ so $\mathbb{1}_{\mathcal{H}_{n-1}} = \mathbb{1}_{\mathcal{H}_{n-2}} - \mathbb{1}_{\tilde{\mathcal{H}}_{n-1}}$. Then, (C.55) yields

$$R_n - R_{n-1} = 2M_{n-1}\gamma_n \xi_n + \gamma_n^2 \xi_n^2 + \gamma_n \psi_n + \gamma_n^2 \theta_n^2 \tag{C.58}$$

so

$$\mathbb{E}[(R_n - R_{n-1}) \mathbb{1}_{\mathcal{H}_{n-1}}] = 2\,\mathbb{E}[\gamma_n M_{n-1} \xi_n \mathbb{1}_{\mathcal{H}_{n-1}}] \tag{C.59a}$$

$$+ \mathbb{E}[\gamma_n^2 \xi_n^2 \mathbb{1}_{\mathcal{H}_{n-1}}] \tag{C.59b}$$

$$+ \mathbb{E}[(\gamma_n \psi_n + \gamma_n^2 \theta_n^2) \mathbb{1}_{\mathcal{H}_{n-1}}] \tag{C.59c}$$

However, since $\mathcal{H}_{n-1}$ and M_{n-1} are both $\mathcal{F}_n$-measurable, we have the following estimates:

1. For the noise term in (C.59a), we have:

$$\mathbb{E}[M_{n-1} \xi_n \mathbb{1}_{\mathcal{H}_{n-1}}] = \mathbb{E}[M_{n-1} \mathbb{1}_{\mathcal{H}_{n-1}} \mathbb{E}[\xi_n \,|\, \mathcal{F}_n]] = 0. \tag{C.60}$$

2. The term (C.59b) is where the reduction to $\mathcal{H}_{n-1}$ kicks in; indeed:

$$\begin{aligned}\mathbb{E}[\xi_n^2 \mathbb{1}_{\mathcal{H}_{n-1}}] &= \mathbb{E}[\mathbb{1}_{\mathcal{H}_{n-1}} \mathbb{E}[|\langle \nabla E(Z_n), U_n\rangle|^2 \,|\, \mathcal{F}_n]] \\ &\leq \mathbb{E}[\mathbb{1}_{\mathcal{H}_{n-1}} \|\nabla E(Z_n)\|^2 \mathbb{E}[\|U_n\|^2 \,|\, \mathcal{F}_n]] && \{\text{by Cauchy–Schwarz}\} \\ &\leq \mathbb{E}[\mathbb{1}_{\mathcal{E}_n} \|\nabla E(Z_n)\|^2 \mathbb{E}[\|U_n\|^2 \,|\, \mathcal{F}_n]] && \{\text{because } \mathcal{H}_{n-1} \subseteq \mathcal{E}_n\} \\ &\leq G^2 \sigma_n^2, && \{\text{by Eq. (4.5b)}\}\end{aligned}$$

where $G^2 = \sup_{z\in\mathcal{K}}\{\|\nabla E(z)\|^2 + \|V(z)\|^2\}$.

3. Finally, for the term (C.59c), we have:

$$\mathbb{E}[\theta_n^2 \mathbb{1}_{\mathcal{H}_{n-1}}] \leq 2\beta\mathbb{E}[\|V(Z_n)\|^2 \mathbb{1}_{\mathcal{E}_n} + \|U_n\|^2] \leq 2\beta(G^2 + \sigma_n^2), \tag{C.61}$$

where we used the fact that $\mathbb{1}_{\mathcal{H}_{n-1}} \leq \mathbb{1}_{\mathcal{E}_n} \leq 1$. Likewise,

$$\mathbb{E}[\psi_n \mathbb{1}_{\mathcal{H}_{n-1}}] \leq GB_n + \gamma_n\beta B_n^2. \tag{C.62}$$

Thus, putting together all of the above, we obtain:

$$\mathbb{E}[(R_n - R_{n-1}) \mathbb{1}_{\mathcal{H}_{n-1}}] \leq \gamma_n GB_n + \gamma_n^2[2\beta G^2 + (2\beta + G^2)\sigma_n^2 + \beta B_n^2]. \tag{C.63}$$

Going back to (C.57), we have $R_{n-1} > \varepsilon$ if $\tilde{\mathcal{H}}_{n-1}$ occurs, so the last term becomes

$$\mathbb{E}[R_{n-1} \mathbb{1}_{\tilde{\mathcal{H}}_{n-1}}] \geq \varepsilon\mathbb{E}[\mathbb{1}_{\tilde{\mathcal{H}}_{n-1}}] = \varepsilon\mathbb{P}(\tilde{\mathcal{H}}_{n-1}). \tag{C.64}$$

Our claim then follows by combining Eqs. (C.57), (C.61), (C.62) and (C.64). ■

Containment probability. Lemma C.1 is the key to showing that Z_n remains close to $\mathcal{S}$ with high probability: we formalize this in a final intermediate result below.

Proposition C.5. *Fix some confidence threshold* $\alpha > 0$. *If* (RM) *is run with sufficiently small* γ_n *satisfying the conditions of Proposition 4.1, then*

$$\mathbb{P}(\mathcal{H}_n \,|\, Z_1 \in \mathcal{U}) \geq 1 - \alpha \quad \textit{for all } n = 1, 2, \ldots \tag{C.65}$$

i.e., Z remains within the basin of attraction $\mathcal{K}$ *of* $\mathcal{S}$ *with probability at least* $1 - \alpha$.

Proof. We begin by bounding the probability of the "bad realization" event $\tilde{\mathcal{H}}_n = \mathcal{H}_{n-1} \setminus \mathcal{H}_n$.

Indeed, if $Z_1 \in \mathcal{U}$, we have:

$$\begin{aligned}\mathbb{P}(\tilde{\mathcal{H}}_n) &= \mathbb{P}(\mathcal{H}_{n-1} \setminus \mathcal{H}_n) = \mathbb{P}(\mathcal{H}_{n-1} \cap \{R_n > \varepsilon\}) \\ &= \mathbb{E}[\mathbb{1}_{\mathcal{H}_{n-1}} \times \mathbb{1}_{\{R_n > \varepsilon\}}] \\ &\leq \mathbb{E}[\mathbb{1}_{\mathcal{H}_{n-1}} \times (R_n/\varepsilon)] \\ &= \mathbb{E}[\tilde{R}_n]/\varepsilon \end{aligned} \tag{C.66}$$

where, in the second-to-last line, we used the fact that $R_n \geq 0$ (so $\mathbb{1}_{\{R_n > \varepsilon\}} \leq R_n/\varepsilon$). Telescoping (C.52) yields

$$\mathbb{E}[\tilde{R}_n] \leq \mathbb{E}[\tilde{R}_0] + G \sum_{k=1}^{n} \gamma_k B_k + \sum_{k=1}^{n} \gamma_k^2 \varrho_k^2 - \varepsilon \sum_{k=1}^{n} \mathbb{P}(\tilde{\mathcal{H}}_{k-1}) \tag{C.67}$$

where we set $\varrho_n^2 = 2\beta G^2 + (2\beta + G^2)\sigma_n^2 + \beta B_n^2$. Hence, combining (C.66) and (C.67) and invoking Assumptions (A5.1) and (A5.2), we get $\sum_{k=1}^{n} \mathbb{P}(\tilde{\mathcal{H}}_k) \leq \frac{1}{\varepsilon} \sum_{k=1}^{n} [\gamma_k G B_k + \gamma_k^2 \varrho_k^2] \leq \Gamma/\varepsilon$ for some $\Gamma > 0$. Now, by choosing γ_n sufficiently small, we can ensure that $\Gamma/\varepsilon < \alpha$; therefore, given that the events $\tilde{\mathcal{H}}_k$ are disjoint for all $k = 1, 2, \ldots$, we get

$$\mathbb{P}\left(\bigcup_{k=1}^{n} \tilde{\mathcal{H}}_k\right) = \sum_{k=1}^{n} \mathbb{P}(\tilde{\mathcal{H}}_k) \leq \alpha \tag{C.68}$$

and hence:

$$\mathbb{P}(\mathcal{H}_n) = \mathbb{P}\left(\bigcap_{k=1}^{n} \tilde{\mathcal{H}}_k^{\mathrm{c}}\right) \geq 1 - \alpha, \tag{C.69}$$

as claimed. ■

Convergence with high probability. We are finally in a position to prove the convergence of generalized RM algorithms:

Proof of Theorem 4.3. By Proposition C.5, if Z_n is initialized within the neighborhood $\mathcal{U}$ defined in (C.50), we have $\mathbb{P}(Z_n \in \mathcal{K} \mid Z_1 \in \mathcal{U}) \geq 1 - \alpha$ (note also that the neighborhood $\mathcal{U}$ is independent of the required confidence level α). Since $\mathcal{K}$ is compact, if $Z_n \in \mathcal{K}$ for all n, we conclude by Theorem 4.1 that the continuous-time interpoloation $Z(t)$ of Z_n is an APT of (MD).

Now, if we write $\mathcal{L} = \bigcap_{t \geq 0} \operatorname{cl}(Z(t, \infty))$ for the limit set of $Z(t)$, we have $\mathcal{K} \cap \mathcal{L} \neq \varnothing$ by the compactness of $\mathcal{K}$ and the fact that $Z_n \in \mathcal{K}$ for all $n \geq 1$; moreover, $\mathcal{L}$ is itself compact as a closed subset of the compact set $\{\Theta_t(z) : 0 \leq t \leq T, z \in \mathcal{K}\}$. Since points in $\mathcal{L} \cap \mathcal{K}$ are a fortiori attracted to $\mathcal{S}$ under (MD) and $\mathcal{L}$ is invariant under (MD), we conclude that $\mathcal{L} \cap \mathcal{S} \neq \varnothing$. However, since $\mathcal{L}$ is internally chain-transitive (by Theorem 4.2) and internally chain-transitive sets do not contain any proper attractors, we conclude that $\mathcal{L} \subseteq \mathcal{S}$. This shows that $Z(t)$ – and, by consequence, Z_n – converges to $\mathcal{S}$, as claimed. ■

C.3 Omitted proofs for Section 4.5

C.3.1 A general criterion for spurious ICT sets in almost bilinear games

We first provide a generic criterion for the existence of spurious ICT sets in almost bilinear games (4.8); cf. Lemma C.2. We then verify that the perturbation $\phi(\boldsymbol{y}) = \frac{1}{2}\boldsymbol{y}^2 - \frac{1}{4}\boldsymbol{y}^4$ employed in Example 4.5.1 indeed satisfies the required conditions.

Lemma C.2. *Let* $\phi(\boldsymbol{y}) = \sum_k a_k \boldsymbol{y}^k$ *be an analytic function such that*

$$\sum_k a_{2k} k h^{2k} \prod_{i=1}^{k} \frac{2i-1}{2i} = 0 \tag{C.70}$$

has a solution with $h > 0$. *Then, for small enough* ε, *there is an ICT set of mean dynamics* (MD) *with objective* $F(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}\boldsymbol{y} + \varepsilon\phi(\boldsymbol{y})$ *such that it does not contain any critical point.*

Proof. Recall the mean dynamics (MD):

$$\dot{z}(t) = V(z(t)).$$

In the case of $F(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}\boldsymbol{y} + \varepsilon\phi(\boldsymbol{y})$, (MD) reads:

$$\begin{cases} \dot{\boldsymbol{x}} = -\boldsymbol{y} \\ \dot{\boldsymbol{y}} = \boldsymbol{x} + \varepsilon\phi'(\boldsymbol{y}) \end{cases}. \tag{C.71}$$

The most important tool of the proof is the *Abelian integral* [CL07]:

$$I(h) := -\oint_{\gamma_h} \phi' \mathrm{d}\boldsymbol{x} \tag{AI}$$

where $h > 0$ is a parameter and γ_h is a family of ovals defined as in (2.3) of [CL07].

Suppose $\phi(\boldsymbol{y}) = a_k \boldsymbol{y}^k$, so that $\phi'(\boldsymbol{y}) = k a_k \boldsymbol{y}^{k-1}$. We choose $\gamma_h = \{z : \|z\| = h\}$. Then, using the polar coordinate representation, we get

$$\begin{aligned} I(h) &= -\oint_{\gamma_h} \phi' \mathrm{d}\boldsymbol{x} \\ &= k a_k \int_0^{2\pi} h^k \sin^k(\theta) \mathrm{d}\theta \\ &= k a_k \cdot \begin{cases} 0 & \text{if } k \text{ is odd,} \\ 2\pi h^k \prod_{i=1}^{\frac{k}{2}} \frac{2i-1}{2i} & \text{if } k \text{ is even.} \end{cases} \end{aligned} \tag{C.72}$$

Since contour integrals are linear in the integrands, when $\phi(\boldsymbol{y}) = \sum_k a_k \boldsymbol{y}^k$ in (AI), we have

$$I(h) = 4\pi \sum_k a_{2k} k h^{2k} \prod_{i=1}^{k} \frac{2i-1}{2i}.$$

Therefore, $I(h) = 0$ if and only if (C.70) holds. By Theorem 2.4 in [CL07], the solution h^* of $I(h^*) = 0$ then implies the existence of a limit cycle in a neighborhood of the oval $\gamma_{h^*} := \{z : \|z\| = h^*\}$. ■

Finally, it is easy to verify that for $\phi(\boldsymbol{y}) = \frac{1}{2}\boldsymbol{y}^2 - \frac{1}{4}\boldsymbol{y}^4$, the condition (C.70) is satisfied with $h^* = \sqrt{\frac{4}{3}}$, thus implying the existence of a spurious ICT set near the neighborhood of $\{z : \|z\| = \sqrt{\frac{4}{3}}\}$.

C.3.2 Proof of spurious ICT sets in Example 4.5.2

We show the existence of two spurious ICT sets in Example 4.5.2.

The mean dynamics (MD) for (4.9) reads:

$$\begin{cases} \dot{\boldsymbol{x}} = -(\boldsymbol{y} - 0.5) - \frac{1}{2}\boldsymbol{x} + 2\boldsymbol{x}^3 - \boldsymbol{x}^5 \\ \dot{\boldsymbol{y}} = \boldsymbol{x} - \frac{1}{2}\boldsymbol{y} + 2\boldsymbol{y}^3 - \boldsymbol{y}^5 \end{cases}. \tag{C.73}$$

Define $r^2 := \boldsymbol{x}^2 + \boldsymbol{y}^2$. Then straightforward calculations show that:

$$\begin{aligned} \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t} r^2 &= \boldsymbol{x}\dot{\boldsymbol{x}} + \boldsymbol{y}\dot{\boldsymbol{y}} \\ &= -\boldsymbol{x}(\boldsymbol{y} - 0.5) - \frac{1}{2}\boldsymbol{x}^2 + 2\boldsymbol{x}^4 - \boldsymbol{x}^6 + \boldsymbol{x}\boldsymbol{y} - \frac{1}{2}\boldsymbol{y}^2 + 2\boldsymbol{y}^4 - \boldsymbol{y}^6 \\ &= 0.5\boldsymbol{x} - \frac{1}{2}r^2 + 2r^4 - r^6 + 3\boldsymbol{x}^4\boldsymbol{y}^2 + 3\boldsymbol{x}^2\boldsymbol{y}^4 - 4\boldsymbol{x}^2\boldsymbol{y}^2 \\ &= 0.5\boldsymbol{x} - \frac{1}{2}r^2 + 2r^4 - r^6 + \boldsymbol{x}^2\boldsymbol{y}^2\left(3r^2 - 4\right). \end{aligned} \tag{C.74}$$

Substituting the value $r^2 = \frac{4}{3}$ into (C.74), we get

$$\begin{aligned} \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t} r^2 &= 0.5\boldsymbol{x} + \frac{1}{2}\cdot\frac{4}{3} + 2\cdot\frac{16}{9} - \frac{64}{27} \\ &= 0.5\boldsymbol{x} + \frac{14}{27} \\ &> 0 \end{aligned}$$

since $|\boldsymbol{x}| \leq \sqrt{\frac{4}{3}}$ on $\{r \geq 0 : r^2 = \frac{4}{3}\}$, whence $\dot{r} > 0$ on $\{r \geq 0 : r^2 = \frac{4}{3}\}$. Likewise, one can check that $\dot{r} < 0$ on $\{r \geq 0 : r^2 = 2\}$, and that there is no stationary point in the region $\mathcal{S} := \{r \geq 0 : \frac{4}{3} \leq r^2 \leq 2\}$. By the Poincaré-Bendixson theorem [Wig03], there exists at least a limit cycle in $\mathcal{S}$.

Finally, it is easy to see that $(\boldsymbol{x}^\star, \boldsymbol{y}^\star) = (0, 0.5)$ is a stable critical point of (4.9). Since the region

$\mathcal{S}$ is trapping, Poincaré's index theorem then dictates that there exists at least another unstable limit cycle inside $\mathcal{S}$, establishing the claim.

C.3.3 Second-order methods in Example 4.5.3 as perturbations

In this section, we discuss how to cast existing second-order methods as an RM scheme with different driving vector fields, and show that their ICT sets are similar to the first-order methods under practical settings.

We will showcase on the *consensus optimization* (ConO):

$$Z_{n+1} = Z_n + \gamma_n (I - \lambda J(Z_n)) V(Z_n) \tag{ConO}$$

where $\lambda > 0$ is the regularization parameter. Recalling the efficient implementation scheme of Hessian-gradient multiplication [Pea94], we make the following assumption on the *stochastic second-order oracles* (SSO): when called at $z = (\boldsymbol{x}, \boldsymbol{y})$ with random seed $\omega' \in \Omega$, an SSO returns a random vector $\mathsf{JV}(z;\omega')$ of the form

$$\mathsf{JV}(z;\omega') = J(z)V(z) + \mathsf{U}'(z;\omega') \tag{SSO}$$

where $\mathsf{U}'(z;\omega')$ is assumed to be unbiased and sub-Gaussian as in (4.2). With these assumptions, one can then proceed exactly as in Appendix C.1.3 for the (SGDA) and (alt-SGDA) cases to show that ConO, and its alternating version, give rise to asymptotic pseudotrajectories of the continuous-time dynamics:

$$\dot{z}(t) = \Big(I - \lambda J(z(t))\Big) V(z(t)).$$

Similarly, one can show (under appropriate assumptions of the oracles) the continuous-time dynamics of *symplectic gradient adjustment* (SGA) is

$$\dot{z}(t) = \left(I - \lambda \left(\frac{J(z(t)) - J(z(t))^\top}{2}\right)\right) V(z(t)).$$

As explained in Example 4.5.3, it is undesirable to set a large number of λ, since then we are essentially treating $\min\max$ and $\max\min$ as the same problem. However, if λ is small, then by continuity, any stable (unstable) ICT set of (MD) remains stable (unstable) under perturbations [Wig03]. We therefore expect the ICT sets of various second-order algorithms in Example 4.5.3 be to similar to that of first-order RM schemes.

C.3.4 Further comparisons

This section includes further comparison of the ICT sets of various algorithms, and show that these existing methods all suffer from the spurious convergence depicted in Section 4.5.

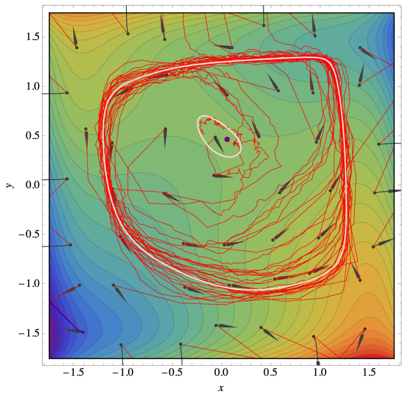


Figure C.1: ConO with $\lambda = 0.2$ applied to (4.9).

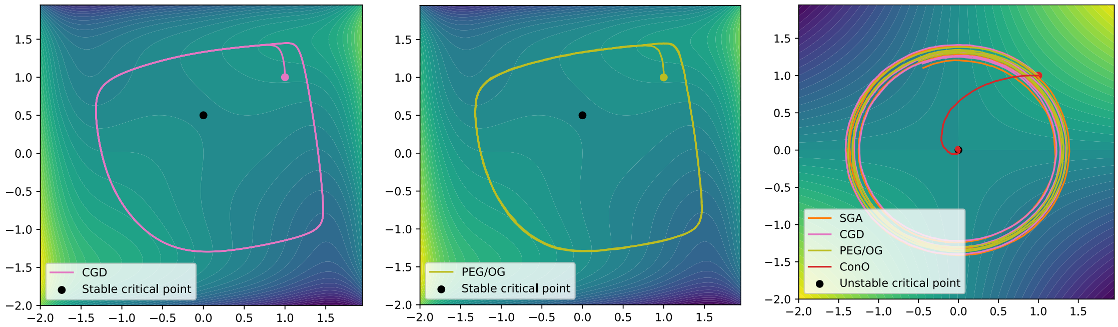


Figure C.2: Spurious limits of min-max optimization algorithms from the same initialization. From left to right: (*a*) CGA for (4.9); (*b*) (OG/PEG) for (4.9); (*c*) Algorithms for (4.8).

First, Fig. C.1 demonstrates that the spurious ICT sets of ConO for (4.9) is similar to that of SGA; cf. Fig. 4.2(c).

Second, we have included yet another second-order method, the *Competitive Gradient Descent* (CGD) [SA19], in Fig. C.2(a). For ease of comparison, we run (OG/PEG) with the same initialization in Fig. C.2(b). As is evident from the figure, both algorithms perform similarly and converge straight to the spurious ICT set.

Finally, we report the bahvior of various algorithms applied to the "almost bilinear game" (4.8). In this case, all algorithms fail to escape the spurious ICT set, with the sole exception of ConO. Intriguingly, ConO converges to the *unstable* critical point. A plausible explanation of this phenomenon is provided by [ALW19], where it is shown that the Hamiltonian descent (HD) converges to critical points for any almost bilinear game. Therefore, it is not surprising that ConO, being a mixture of SGDA and HD, also enjoys similar guarantees. Such a convergence is nonetheless highly undesirable in our example, echoing the concern that gradient penalty schemes cannot distinguish (local) $\min\max$ from $\max\min$.

D Appendix for Chapter 5

D.1 A framework for infinite-dimensional mirror descent

D.1.1 A note on the regularity

It is known that the (negative) Shannon entropy is *not* Fréchet differentiable in general. However, below we show that the Fréchet derive can be well-defined if we restrict the probability measures to within the set

$$\begin{aligned}\mathcal{M}(\mathcal{Z}) := \{&\text{all probability measures on } \mathcal{Z} \text{ that admit densities w.r.t. the Lebesgue measure,}\\ &\text{and the density is continuous and positive almost everywhere on } \mathcal{Z}\}. \end{aligned} \tag{D.1}$$

We will also restrict the set of functions to be bounded and integrable:

$$\mathcal{F}(\mathcal{Z}) := \left\{\text{all bounded continuous functions } f \text{ on } \mathcal{Z} \text{ such that } \int e^{-f} < \infty \right\}. \tag{D.2}$$

These assumptions on the probability measures and functions are sufficient for most practical applications.

It is important to notice that $\mu \in \mathcal{M}(\mathcal{Z})$ and $h \in \mathcal{F}(\mathcal{Z})$ implies $\mu' = \mathrm{MD}_\eta\left(\mu, h\right) \in \mathcal{M}(\mathcal{Z})$; this readily follows from the formula (5.7).

D.1.2 Properties of entropic mirror map

The total variation of a (possibly non-probability) measure $\mu \in \mathcal{M}(\mathcal{Z})$ is defined as [Hal13]

$$\|\mu\|_{\mathrm{TV}} = \sup_{\|h\|_{\mathbb{L}^\infty} \le 1} \int h \mathrm{d}\mu = \sup_{\|h\|_{\mathbb{L}^\infty} \le 1} \left\langle \mu, h \right\rangle.$$

Recall the standard topology induced by $\|\cdot\|_{\mathrm{TV}}$ and $\|\cdot\|_{\mathbb{L}^\infty}$ for measures and functions [Hal13], respectively. Whenever we speak about continuity or differentiability below, it is understood to be w.r.t. to the standard topology. Notice also that the G operator defined in (5.5) is bounded if the discriminator $f_{\boldsymbol{w}}$ is bounded, and hence continuous [Hal13].

We depart from the fundamental *Gibbs Variational Principle*, which dates back to the earliest work of statistical mechanics [Gib02]. For two probability measures μ, μ', denote their relative entropy by (the reason for this notation will become clear in (D.8))

$$D_\Phi(\mu, \mu') := \int_{\mathcal{Z}} \mathrm{d}\mu \log \frac{\mathrm{d}\mu}{\mathrm{d}\mu'}.$$

By the definition of $\mathcal{M}(\mathcal{Z})$, it is clear that the relative entropy is well-defined for any $\mu, \mu' \in \mathcal{M}(\mathcal{Z})$.

Theorem D.1 (*Gibbs Variation Principle). Let $h \in \mathcal{F}(\mathcal{Z})$ and $\mu' \in \mathcal{M}(\mathcal{Z})$ be a reference measure. Then*

$$\log \int_{\mathcal{Z}} e^h \mathrm{d}\mu' = \sup_{\mu \in \mathcal{M}(\mathcal{Z})} \langle \mu, h \rangle - D_\Phi(\mu, \mu'), \tag{D.3}$$

and equality is achieved by $\mathrm{d}\mu^\star = \frac{e^h \mathrm{d}\mu'}{\int_{\mathcal{Z}} e^h \mathrm{d}\mu'}$.

Part of the following theorem is folklore in the mathematics and learning community. However, to the best of our knowledge, the relation to the entropic MD has not been systematically studied before, as we now do.

Theorem D.2. *For a probability measure* $\mathrm{d}\mu = \rho \mathrm{d}\boldsymbol{z}$, *let* $\Phi(\mu) = \int \rho \log \rho \mathrm{d}\boldsymbol{z}$ *be the negative Shannon entropy, and let* $\Phi^\star(h) = \log \int_{\mathcal{Z}} e^h \mathrm{d}\boldsymbol{z}$. *Then*

1. $\Phi^\star$ *is the Fenchel conjugate of* Φ*:*

$$\Phi^\star(h) = \sup_{\mu \in \mathcal{M}(\mathcal{Z})} \langle \mu, h \rangle - \Phi(\mu); \tag{D.4}$$

$$\Phi(\mu) = \sup_{h \in \mathcal{F}(\mathcal{Z})} \langle \mu, h \rangle - \Phi^\star(h). \tag{D.5}$$

2. *The derivatives admit the expression*

$$\mathrm{d}\Phi(\mu) = 1 + \log \rho = \operatorname*{argmax}_{h \in \mathcal{F}(\mathcal{Z})} \langle \mu, h \rangle - \Phi^\star(h); \tag{D.6}$$

$$\mathrm{d}\Phi^\star(h) = \frac{e^h \mathrm{d}\boldsymbol{z}}{\int_{\mathcal{Z}} e^h \mathrm{d}\boldsymbol{z}} = \operatorname*{argmax}_{\mu \in \mathcal{M}(\mathcal{Z})} \langle \mu, h \rangle - \Phi(\mu). \tag{D.7}$$

3. *The Bregman divergence of* Φ *is the relative entropy:*

$$D_\Phi(\mu, \mu') = \Phi(\mu) - \Phi(\mu') - \langle \mu - \mu', \mathrm{d}\Phi(\mu') \rangle = \int_{\mathcal{Z}} \mathrm{d}\mu \log \frac{\mathrm{d}\mu}{\mathrm{d}\mu'}. \tag{D.8}$$

4. *Φ is 4-strongly convex with respect to the total variation norm: For all $\lambda \in (0,1)$,*

$$\Phi(\lambda\mu + (1-\lambda)\mu') \le \lambda\Phi(\mu) + (1-\lambda)\Phi(\mu') - \frac{1}{2}\cdot 4\lambda(1-\lambda)\|\mu-\mu'\|_{\mathrm{TV}}^2. \tag{D.9}$$

5. *The following duality relation holds: For any constant C, we have*

$$\forall \mu,\mu' \in \mathcal{M}(\mathcal{Z}), \quad D_\Phi(\mu,\mu') = D_{\Phi^\star}\left(\mathrm{d}\Phi(\mu'),\mathrm{d}\Phi(\mu)\right) = D_{\Phi^\star}\left(\mathrm{d}\Phi(\mu')+C,\mathrm{d}\Phi(\mu)\right). \tag{D.10}$$

6. *$\Phi^\star$ is $\frac{1}{4}$-smooth with respect to $\|\cdot\|_{\mathbb{L}^\infty}$:*

$$\forall h,h' \in \mathcal{F}(\mathcal{Z}), \quad \left\|\mathrm{d}\Phi^\star(h) - \mathrm{d}\Phi^\star(h')\right\|_{\mathrm{TV}} \le \frac{1}{4}\left\|h-h'\right\|_{\mathbb{L}^\infty}. \tag{D.11}$$

7. *Alternative to* (D.11)*, we have the equivalent characterization of $\Phi^\star$:*

$$\forall h,h' \in \mathcal{F}(\mathcal{Z}), \quad \Phi^\star(h) \le \Phi^\star(h') + \left\langle \mathrm{d}\Phi^\star(h'), h-h'\right\rangle + \frac{1}{2}\cdot\frac{1}{4}\left\|h-h'\right\|_{\mathbb{L}^\infty}^2. \tag{D.12}$$

8. *Similar to* (D.10)*, we have*

$$\forall h,h', \quad D_{\Phi^\star}(h,h') = D_\Phi(\mathrm{d}\Phi^\star(h'),\mathrm{d}\Phi^\star(h)). \tag{D.13}$$

9. *The following three-point identity holds for all $\mu,\mu',\mu'' \in \mathcal{M}(\mathcal{Z})$:*

$$\left\langle \mu''-\mu, \mathrm{d}\Phi(\mu') - \mathrm{d}\Phi(\mu)\right\rangle = D_\Phi(\mu,\mu') + D_\Phi(\mu'',\mu) - D_\Phi(\mu'',\mu'). \tag{D.14}$$

10. *Let the Mirror Descent iterate be defined as in* (5.7)*. Then the following statements are equivalent:*
    - *(a)* $\mu_+ = \mathrm{MD}_\eta\left(\mu,h\right)$.
    - *(b) There exists a constant C such that* $\mathrm{d}\Phi(\mu_+) = \mathrm{d}\Phi(\mu) - \eta h + C$.

    In particular, for any $\mu',\mu'' \in \mathcal{M}(\mathcal{Z})$ we have

$$\mathit{Let}\left\langle \mu'-\mu'', \eta h\right\rangle = \left\langle \mu'-\mu'', \mathrm{d}\Phi(\mu) - \mathrm{d}\Phi(\mu_+)\right\rangle. \tag{D.15}$$

Proof.

1. Equation (D.4) is simply the Gibbs variational principle (D.3) with $\mathrm{d}\mu \leftarrow \mathrm{d}\boldsymbol{z}$.

   By (D.4), we know that

$$\forall h \in \mathcal{F}(\mathcal{Z}), \quad \Phi(\mu) \ge \left\langle \mu, h\right\rangle - \log\int_{\mathcal{Z}} e^h \mathrm{d}\boldsymbol{z}. \tag{D.16}$$

   But for $\mathrm{d}\mu = \rho\mathrm{d}\boldsymbol{z}$, the function $h := 1 + \log\rho$ saturates the equality in (D.16).

2. We prove a more general result on the Bregman divergence D_Φ in (D.17) below.

   Let $\mathrm{d}\mu = \rho \mathrm{d}\boldsymbol{z}, \mathrm{d}\mu' = \rho' \mathrm{d}\boldsymbol{z}$, and $\mathrm{d}\mu'' = \rho'' \mathrm{d}\boldsymbol{z} \in \mathcal{M}(\mathcal{Z})$. Let $\epsilon > 0$ be small enough such that $(\rho + \epsilon\rho'')\mathrm{d}\boldsymbol{z}$ is absolutely continuous with respect to $\mathrm{d}\mu'$; note that this is possible because μ, μ', and $\mu'' \in \mathcal{M}(\mathcal{Z})$. We compute

$$\begin{aligned} D_\Phi(\rho + \epsilon\rho'', \rho') &= \int_{\mathcal{Z}} (\rho + \epsilon\rho'') \log \frac{\rho + \epsilon\rho''}{\rho'} \\ &= \int_{\mathcal{Z}} \rho \log \frac{\rho}{\rho'} + \int_{\mathcal{Z}} \rho \log\left(1 + \epsilon \frac{\rho''}{\rho}\right) + \epsilon \int_{\mathcal{Z}} \rho'' \log \frac{\rho}{\rho'} + \epsilon \int_{\mathcal{Z}} \rho'' \log\left(1 + \epsilon \frac{\rho''}{\rho}\right) \\ &\overset{\text{(i)}}{=} \int_{\mathcal{Z}} \rho \log \frac{\rho}{\rho'} + \epsilon \int_{\mathcal{Z}} \rho'' + \epsilon \int_{\mathcal{Z}} \rho'' \log \frac{\rho}{\rho'} + \epsilon^2 \int_{\mathcal{Z}} \frac{\rho''^2}{\rho} + o(\epsilon) \\ &= D_\Phi(\rho, \rho') + \epsilon \int_{\mathcal{Z}} \rho'' \left(1 + \log \frac{\rho}{\rho'}\right) + o(\epsilon), \end{aligned}$$

   where (i) uses $\log(1+t) = t + o(t)$ as $t \to 0$. In short, for all $\mu', \mu'' \in \mathcal{M}(\mathcal{Z})$,

$$\mathrm{d}_\mu D_\Phi(\mu, \mu')(\mu'') = \left\langle \mu'', 1 + \log \frac{\rho}{\rho'} \right\rangle \tag{D.17}$$

   which means $\mathrm{d}_\mu D_\Phi(\mu, \mu') = 1 + \log \frac{\rho}{\rho'}$. The formula (D.6) is the special case when $\mathrm{d}\mu' \leftarrow \mathrm{d}\boldsymbol{z}$.

   We now turn to (D.7). For every $h \in \mathcal{F}(\mathcal{Z})$, we need to show that the following holds for every $h' \in \mathcal{F}(\mathcal{Z})$:

$$\Phi^\star(h + \epsilon h') - \Phi^\star(h) = \log \int_{\mathcal{Z}} e^{h + \epsilon h'} \mathrm{d}\boldsymbol{z} - \log \int_{\mathcal{Z}} e^h \mathrm{d}\boldsymbol{z} = \epsilon \int_{\mathcal{Z}} h' \frac{e^h}{\int_{\mathcal{Z}} e^h} \mathrm{d}\boldsymbol{z} + o(\epsilon). \tag{D.18}$$

   Define an auxiliary function

$$T(\epsilon) := \log \int_{\mathcal{Z}} \frac{e^h}{\int_{\mathcal{Z}} e^h} e^{\epsilon h'} \mathrm{d}\boldsymbol{z}.$$

   Notice that $T(0) = 0$ and T is smooth as a function of ϵ. Thus, by the Intermediate Value Theorem,

$$\begin{aligned} \Phi^\star(h + \epsilon h') - \Phi^\star(h) &= T(\epsilon) - T(0) \\ &= (\epsilon - 0) \cdot \frac{\mathrm{d}}{\mathrm{d}\epsilon} T(\cdot)\Big|_{\epsilon'} \end{aligned}$$

   for some $\epsilon' \in [0, \epsilon]$. A direct computation shows

$$\frac{d}{d\epsilon} T(\cdot)\Big|_{\epsilon'} = \int_{\mathcal{Z}} h' \frac{e^{h + \epsilon' h'}}{\int_{\mathcal{Z}} e^{h + \epsilon' h'}} \mathrm{d}\boldsymbol{z}.$$

   Hence it suffices to prove $\frac{e^{h+\epsilon' h'}}{\int_{\mathcal{Z}} e^{h+\epsilon' h'}} = \frac{e^h}{\int_{\mathcal{Z}} e^h} + o(1)$ in ϵ. To this end, let $C = \sup |h'| < \infty$.

Then

$$\frac{e^h}{\int_{\mathcal{Z}} e^h} e^{-2\epsilon' C} \le \frac{e^{h+\epsilon' h'}}{\int_{\mathcal{Z}} e^{h+\epsilon' h'}} \le \frac{e^h}{\int_{\mathcal{Z}} e^h} e^{2\epsilon' C}.$$

It remains to use $e^t = 1 + t + o(t)$ and $\epsilon' \le \epsilon$.

3. Let $\mathrm{d}\mu = \rho \mathrm{d}\boldsymbol{z}$ and $\mathrm{d}\mu' = \rho' \mathrm{d}\boldsymbol{z}$. We compute

$$\begin{aligned} D_\Phi(\mu,\mu') &= \Phi(\mu) - \Phi(\mu') - \langle \mu - \mu', \mathrm{d}\Phi(\mu')\rangle \\ &= \int_{\mathcal{Z}} \rho \log \rho \mathrm{d}\boldsymbol{z} - \int_{\mathcal{Z}} \rho' \log \rho' \mathrm{d}\boldsymbol{z} - \langle \mu - \mu', 1 + \log \rho' \rangle && \text{by (D.6)} \\ &= \int_{\mathcal{Z}} \rho \log \frac{\rho}{\rho'} \mathrm{d}\boldsymbol{z} \\ &= \int_{\mathcal{Z}} \mathrm{d}\mu \log \frac{\mathrm{d}\mu}{\mathrm{d}\mu'}. \end{aligned}$$

4. Define $\mu_\lambda = \lambda\mu + (1-\lambda)\mu'$. By (D.8) and the classical Pinsker's inequality [Gra11], we have

$$\Phi(\mu) \ge \Phi(\mu_\lambda) + \left\langle (1-\lambda)(\mu-\mu'), \mathrm{d}\Phi(\mu_\lambda)\right\rangle + 2\|(1-\lambda)(\mu-\mu')\|_{\mathrm{TV}}^2, \tag{D.19}$$

$$\Phi(\mu') \ge \Phi(\mu_\lambda) + \left\langle \lambda(\mu'-\mu), \mathrm{d}\Phi(\mu_\lambda)\right\rangle + 2\|\lambda(\mu-\mu')\|_{\mathrm{TV}}^2. \tag{D.20}$$

Equation (D.9) follows by multiplying with λ and $1-\lambda$ respectively and summing the two inequalities up.

5. Let $\mu = \rho \mathrm{d}\boldsymbol{z}$ and $\mu' = \rho' \mathrm{d}\boldsymbol{z}$. Then, by the definition of Bregman divergence and (D.6), (D.7),

$$\begin{aligned} D_{\Phi^\star}(\mathrm{d}\Phi(\mu'), \mathrm{d}\Phi(\mu)) &= \Phi^\star(\mathrm{d}\Phi(\mu')) - \Phi^\star(\mathrm{d}\Phi(\mu)) - \left\langle \frac{e^{1+\log\rho}\mathrm{d}\boldsymbol{z}}{\int_{\mathcal{Z}} e^{1+\log\rho}}, 1 + \log\rho' - 1 - \log\rho \right\rangle \\ &= \log \int_{\mathcal{Z}} e^{1+\log\rho'} - \log \int_{\mathcal{Z}} e^{1+\log\rho} + \int_{\mathcal{Z}} \rho \log \frac{\rho}{\rho'} \\ &= \int_{\mathcal{Z}} \rho \log \frac{\rho}{\rho'} = D_\Phi(\mu, \mu') \end{aligned}$$

since $\int_{\mathcal{Z}} \rho \mathrm{d}\boldsymbol{z} = \int_{\mathcal{Z}} \rho' \mathrm{d}\boldsymbol{z} = 1$. This proves the first equality.

For the second equality, we write

$$\begin{aligned} D_{\Phi^\star}(\mathrm{d}\Phi(\mu') + C, \mathrm{d}\Phi(\mu)) &= \Phi^\star(\mathrm{d}\Phi(\mu') + C) - \Phi^\star(\mathrm{d}\Phi(\mu)) - \left\langle \frac{e^{1+\log\rho}\mathrm{d}\boldsymbol{z}}{\int_{\mathcal{Z}} e^{1+\log\rho}}, 1 + \log\rho' + C - 1 - \log\rho \right\rangle \\ &= \log \int_{\mathcal{Z}} e^{1+\log\rho'+C} - \log \int_{\mathcal{Z}} e^{1+\log\rho} + \int_{\mathcal{Z}} \rho \log \frac{\rho}{\rho'} - C \\ &= \int_{\mathcal{Z}} \rho \log \frac{\rho}{\rho'} \\ &= D_\Phi(\mu, \mu') = D_{\Phi^\star}(\mathrm{d}\Phi(\mu'), \mathrm{d}\Phi(\mu)) \end{aligned}$$

where we have used the first equality in the last step.

6. Let $\mu_h = \mathrm{d}\Phi^\star(h)$, $\mu_{h'} = \mathrm{d}\Phi^\star(h')$, and $\mu_\lambda = \lambda\mu_h + (1-\lambda)\mu_{h'}$ for some $\lambda \in (0,1)$. By Pinsker's inequality and (D.8), we have

$$\Phi(\mu_\lambda) \geq \Phi(\mu_h) + \langle \mu_\lambda - \mu_h, \mathrm{d}\Phi(\mu_h)\rangle + 2\|\mu_\lambda - \mu_h\|_{\mathrm{TV}}^2, \tag{D.21}$$

$$\Phi(\mu_\lambda) \geq \Phi(\mu_{h'}) + \langle \mu_\lambda - \mu_{h'}, \mathrm{d}\Phi(\mu_{h'})\rangle + 2\|\mu_\lambda - \mu_{h'}\|_{\mathrm{TV}}^2. \tag{D.22}$$

Now, notice that

$$\begin{aligned}\langle \mu_\lambda - \mu_h, \mathrm{d}\Phi(\mu_h)\rangle &= \langle \mu_\lambda - \mu_h, \mathrm{d}\Phi(\mathrm{d}\Phi^\star(h))\rangle \\ &= \left\langle \mu_\lambda - \mu_h, \mathrm{d}\Phi\left(\frac{e^h \mathrm{d}\mathbf{z}}{\int_{\mathcal{Z}} e^h}\right)\right\rangle && \text{by (D.7)} \\ &= \left\langle \mu_\lambda - \mu_h, 1 + h - \log\int_{\mathcal{Z}} e^h\right\rangle && \text{by (D.6)} \\ &= \langle \mu_\lambda - \mu_h, h\rangle\end{aligned}$$

and, similarly, we have $\langle \mu_\lambda - \mu_{h'}, \mathrm{d}\Phi(\mu_{h'})\rangle = \langle \mu_\lambda - \mu_{h'}, h'\rangle$. Multiplying (D.21) by λ and (D.22) by $1-\lambda$, summing the two up, and using the above equalities, we get

$$\Phi(\mu_\lambda) - \Big(\lambda\Phi(\mu_h) + (1-\lambda)\Phi(\mu_{h'})\Big) + \lambda(1-\lambda)\langle \mu_h - \mu_{h'}, h - h'\rangle \geq 2\lambda(1-\lambda)\left\|\mu_h - \mu_{h'}\right\|_{\mathrm{TV}}^2.$$

By (D.9), we know that

$$\Phi(\mu_\lambda) - \Big(\lambda\Phi(\mu_h) + (1-\lambda)F(\mu_{h'})\Big) \leq -2\lambda(1-\lambda)\left\|\mu_h - \mu_{h'}\right\|_{\mathrm{TV}}^2.$$

Moreover, by definition of the total variation norm, it is clear that

$$\langle \mu_h - \mu_{h'}, h - h'\rangle \leq \left\|\mu_h - \mu_{h'}\right\|_{\mathrm{TV}} \left\|h - h'\right\|_{\mathbb{L}^\infty}. \tag{D.23}$$

Combing the last three inequalities gives (D.11).

7. Let K be a positive integer and $k \in \{0,1,2,\dots,K\}$. Set $\lambda_k = \frac{k}{K}$ and $h'' = h - h'$. Then

$$\begin{aligned}\Phi^\star(h) - \Phi^\star(h') &= \Phi^\star(h' + \lambda_K h'') - \Phi^\star(h' + \lambda_0 h'') \\ &= \sum_{k=0}^{K-1}\Big(\Phi^\star(h' + \lambda_{k+1}h'') - \Phi^\star(h' + \lambda_k h'')\Big).\end{aligned} \tag{D.24}$$

By convexity of $\Phi^\star$, we have

$$\begin{aligned}\Phi^\star(h' + \lambda_{k+1}h'') - \Phi^\star(h' + \lambda_k h'') &\leq \langle \mathrm{d}\Phi^\star(h' + \lambda_{k+1}h''), (\lambda_{k+1} - \lambda_k)h''\rangle \\ &= \frac{1}{K}\langle \mathrm{d}\Phi^\star(h' + \lambda_{k+1}h''), h''\rangle.\end{aligned} \tag{D.25}$$

By (D.23) and (D.11), we may further upper bound (D.25) as

$$\begin{aligned}\Phi^\star(h'+\lambda_{k+1}h'')-\Phi^\star(h'+\lambda_k h'') &\le \frac{1}{K}\Big(\big\langle \mathrm{d}\Phi^\star(h'),h''\big\rangle+\big\langle \mathrm{d}\Phi^\star(h'+\lambda_{k+1}h'')-\mathrm{d}\Phi^\star(h'),h''\big\rangle\Big)\\ &\le \frac{1}{K}\Big(\big\langle \mathrm{d}\Phi^\star(h'),h''\big\rangle+\big\|\mathrm{d}\Phi^\star(h'+\lambda_{k+1}h'')-\mathrm{d}\Phi^\star(h')\big\|_{\mathrm{TV}}\left\|h''\right\|_{\mathbb{L}^\infty}\Big)\\ &\le \frac{1}{K}\Big(\big\langle \mathrm{d}\Phi^\star(h'),h''\big\rangle+\frac{\lambda_{k+1}}{4}\left\|h''\right\|_{\mathbb{L}^\infty}^2\Big).\end{aligned}\tag{D.26}$$

Summing up (D.26) over k, we get, in view of (D.24),

$$\begin{aligned}\Phi^\star(h)-\Phi^\star(h') &\le \big\langle \mathrm{d}\Phi^\star(h'),h''\big\rangle+\frac{1}{4}\left\|h''\right\|_{\mathbb{L}^\infty}^2\sum_{k=0}^{K-1}\lambda_{k+1}\\ &= \big\langle \mathrm{d}\Phi^\star(h'),h''\big\rangle+\frac{1}{4}\cdot\frac{K+1}{2K}\left\|h''\right\|_{\mathbb{L}^\infty}^2.\end{aligned}\tag{D.27}$$

Since K is arbitrary, we may take $K\to\infty$ in (D.27), which is (D.12).

8. Straightforward calculation shows

$$D_{\Phi^\star}(h,h')=\log\int_{\mathcal{Z}}e^h-\log\int_{\mathcal{Z}}e^{h'}-\int_{\mathcal{Z}}\frac{e^{h'}}{\int e^{h'}}\left(h-h'\right).$$

On the other hand, by definition of the Bregman divergence and (D.6), (D.7), we have

$$\begin{aligned}D_\Phi(\mathrm{d}\Phi^\star(h'),\mathrm{d}\Phi^\star(h)) &= \int_{\mathcal{Z}}\frac{e^{h'}}{\int_{\mathcal{Z}}e^{h'}}h'-\log\int_{\mathcal{Z}}e^{h'}-\int_{\mathcal{Z}}\frac{e^h}{\int_{\mathcal{Z}}e^h}h+\log\int_{\mathcal{Z}}e^h\\ &\qquad -\int_{\mathcal{Z}}\left(1+h-\log\int_{\mathcal{Z}}e^h\right)\left(\frac{e^{h'}}{\int_{\mathcal{Z}}e^{h'}}-\frac{e^h}{\int_{\mathcal{Z}}e^h}\right)\\ &= \int_{\mathcal{Z}}\frac{e^{h'}}{\int e^{h'}}\left(h'-h\right)-\log\int_{\mathcal{Z}}e^{h'}+\log\int_{\mathcal{Z}}e^h\\ &= \Phi^\star(h)-\Phi^\star(h')-\big\langle \mathrm{d}\Phi^\star(h'),h-h'\big\rangle\\ &= D_{\Phi^\star}(h,h').\end{aligned}$$

9. By definition of the Bregman divergence, we have

$$\begin{aligned}D_\Phi(\mu,\mu') &= \Phi(\mu)-\Phi(\mu')-\big\langle \mu-\mu',\mathrm{d}\Phi(\mu')\big\rangle,\\ D_\Phi(\mu'',\mu) &= \Phi(\mu'')-\Phi(\mu)-\big\langle \mu''-\mu,\mathrm{d}\Phi(\mu)\big\rangle,\\ D_\Phi(\mu'',\mu') &= \Phi(\mu'')-\Phi(\mu')-\big\langle \mu''-\mu',\mathrm{d}\Phi(\mu')\big\rangle.\end{aligned}$$

Equation (D.14) then follows by straightforward calculations.

10. First, let $\mu_+=\mathrm{MD}_\eta\left(\mu,h\right)$. Then if $\mu_+=\rho_+\mathrm{d}\boldsymbol{z}$ and $\mu=\rho\mathrm{d}\boldsymbol{z}$, then (5.7) implies

$$\rho_+=\frac{\rho e^{-\eta h}}{\int_{\mathcal{Z}}\rho e^{-\eta h}}.$$

By (D.6), we therefore have

$$\begin{aligned}\mathrm{d}\Phi(\mu_+) &= 1+\log\rho_+\\ &= 1+\log\rho-\eta h-\log\int_{\mathcal{Z}}\rho e^{-\eta h}\end{aligned}$$

whence (D.15) holds with $C=-\log\int_{\mathcal{Z}}\rho e^{-\eta h}$.

Conversely, assume that $\mathrm{d}\Phi(\mu_+)=\mathrm{d}\Phi(\mu)-\eta h+C$ for some constant C, and apply $\mathrm{d}\Phi^\star$ to both sides. The left-hand side becomes

$$\begin{aligned}\mathrm{d}\Phi^\star\Big(\mathrm{d}\Phi(\mu_+)\Big) &= \mathrm{d}\Phi^\star(1+\log\rho_+)\\ &= \frac{\rho_+\mathrm{d}\boldsymbol{z}}{\int\rho_+\mathrm{d}\boldsymbol{z}}=\rho_+\mathrm{d}\boldsymbol{z}=\mathrm{d}\mu_+,\end{aligned}$$

where as the formula (D.7) implies that

$$\begin{aligned}\mathrm{d}\Phi^\star\big(\mathrm{d}\Phi(\mu)-\eta h+C\big) &= \frac{e^{1+\log\rho-\eta h+C}}{\int_{\mathcal{Z}}e^{1+\log\rho-\eta h+C}}\mathrm{d}\boldsymbol{z}\\ &= \frac{\rho e^{-\eta h}\mathrm{d}\boldsymbol{z}}{\int_{\mathcal{Z}}\rho e^{-\eta h}}\\ &= \frac{e^{-\eta h}\mathrm{d}\mu}{\int_{\mathcal{Z}}e^{-\eta h}\mathrm{d}\mu}.\end{aligned}$$

Combining the two equalities gives $\mathrm{d}\mu_+=\frac{e^{-\eta h}\mathrm{d}\mu}{\int_{\mathcal{Z}}e^{-\eta h}\mathrm{d}\mu}$ which exactly means $\mu_+=\mathrm{MD}_\eta\left(\mu,h\right)$.

■

D.2 Convergence rates for infinite-dimensional prox methods

D.2.1 Rigorous Statements

For Algorithms 7–8, we have the following guarantees:

Theorem D.3 (Convergence Rates). *Let* $\Phi(\mu)=\int\mathrm{d}\mu\log\frac{\mathrm{d}\mu}{\mathrm{d}\boldsymbol{z}}$. *Let* M *be a constant such that* $\max\left[\left\|-g+G\nu\right\|_{\mathbb{L}^\infty},\left\|G^\dagger\mu\right\|_{\mathbb{L}^\infty}\right]\le M$, *and* L *be such that* $\left\|G(\nu-\nu')\right\|_{\mathbb{L}^\infty}\le L\left\|\nu-\nu'\right\|_{\mathrm{TV}}$ *and* $\left\|G^\dagger(\mu-\mu')\right\|_{\mathbb{L}^\infty}\le L\left\|\mu-\mu'\right\|_{\mathrm{TV}}$. *Let* $D(\cdot,\cdot)$ *be the relative entropy, and denote by* $D_0:=D(\mu_{\mathrm{NE}},\mu_1)+D(\nu_{\mathrm{NE}},\nu_1)$ *the initial distance to the mixed NE. Then*

1. *Assume that we have access to the deterministic derivatives* $\left\{-G^\dagger\mu_t\right\}_{t=1}^T$ *and* $\left\{g-G\nu\right\}_{t=1}^T$. *Then Algorithm 7 achieves* $O\left(T^{-\frac{1}{2}}\right)$*-NE with* $\eta=\frac{2}{M}\sqrt{\frac{D_0}{T}}$, *and Algorithm 8 achieves* $O\left(T^{-1}\right)$*-NE with* $\eta=\frac{4}{L}$.
2. *Assume that we have access to stochastic derivatives* $\left\{-\hat{G}^\dagger\mu_t\right\}_{t=1}^T$ *and* $\left\{\hat{g}-\hat{G}\nu\right\}_{t=1}^T$ *such that* $\max\left[\mathbb{E}\left\|-\hat{g}+\hat{G}\nu\right\|_{\mathbb{L}^\infty},\mathbb{E}\left\|\hat{G}^\dagger\mu\right\|_{\mathbb{L}^\infty}\right]\le M'$, *and the variance is upper bounded by* σ^2.

Assume also that the bias of stochastic derivatives satisfies

$$\max\left[\left\|\mathbb{E}[-\hat{g}+\hat{G}v]+g-Gv\right\|_{\mathbb{L}^\infty},\left\|\mathbb{E}[\hat{G}^\dagger\mu]-G^\dagger\mu\right\|_{\mathbb{L}^\infty}\right]\le\tau.$$

Then Algorithm 7 with stochastic derivatives achieves $O\left(T^{-\frac{1}{2}}\right)$*-NE in expectation with* $\eta=\sqrt{\frac{D_0}{T\left(4\tau+\frac{M'}{4}\right)}}$, *and Algorithm 8 with stochastic derivatives achieves* $\left(O\left(T^{-\frac{1}{2}}\right)+O(\tau)\right)$*-NE in expectation with* $\eta=\min\left[\frac{4}{\sqrt{3}L},\sqrt{\frac{2D_0}{3T\sigma^2}}\right]$.

D.2.2 Proof of convergence rates for infinite-dimensional mirror descent

Mirror descent, deterministic derivatives

By the definition of the algorithm, (D.15), and the three-point identity (D.14), we have, for any $\mu\in\mathcal{M}(\mathcal{W})$,

$$\begin{aligned}\left\langle\mu_t-\mu,-g+Gv_t\right\rangle&=\frac{1}{\eta}\left\langle\mu_t-\mu,\mathrm{d}\Phi(\mu_t)-\mathrm{d}\Phi(\mu_{t+1})\right\rangle\\&=\frac{1}{\eta}\Big(D_\Phi(\mu,\mu_t)-D_\Phi(\mu,\mu_{t+1})+D_\Phi(\mu_t,\mu_{t+1})\Big).\end{aligned}\tag{D.28}$$

By Theorem D.2.10, there exists a constant C_t such that

$$\mathrm{d}\Phi(\mu_{t+1})=\mathrm{d}\Phi(\mu_t)-\eta\left(-g+Gv_t\right)+C_t.\tag{D.29}$$

Using (D.10), we see that

$$\begin{aligned}D_\Phi(\mu_t,\mu_{t+1})&=D_{\Phi^\star}(\mathrm{d}\Phi(\mu_{t+1}),\mathrm{d}\Phi(\mu_t))\\&=D_{\Phi^\star}\Big(\mathrm{d}\Phi(\mu_{t+1})-C_t,\mathrm{d}\Phi(\mu_t)\Big)\\&\le\frac{1}{8}\left\|\mathrm{d}\Phi(\mu_{t+1})-C_t-\mathrm{d}\Phi(\mu_t)\right\|_{\mathbb{L}^\infty}^2&&\text{by (D.12)}\\&=\frac{\eta^2}{8}\left\|-g+Gv_t\right\|_{\mathbb{L}^\infty}^2&&\text{by (D.29)}\\&\le\frac{\eta^2M^2}{8}.\end{aligned}$$

Consequently, we have

$$\begin{aligned}\sum_{t=1}^T\left\langle\mu_t-\mu,-g+Gv_t\right\rangle&=\sum_{t=1}^T\frac{1}{\eta}\Big(D_\Phi(\mu,\mu_t)-D_\Phi(\mu,\mu_{t+1})+D_\Phi(\mu_t,\mu_{t+1})\Big)\\&\le\frac{D_\Phi(\mu,\mu_1)}{\eta}+\frac{\eta M^2T}{8}.\end{aligned}\tag{D.30}$$

Exactly the same argument applied to ν_t's yields, for any $\nu \in \mathcal{M}(\Theta)$,

$$\sum_{t=1}^{T} \left\langle \nu_t - \nu, -G^\dagger \mu_t \right\rangle \le \frac{D_\Phi(\nu, \nu_1)}{\eta} + \frac{\eta M^2 T}{8}. \tag{D.31}$$

Summing up (D.30) and (D.31), substituting $\mu \leftarrow \mu_{\mathrm{NE}}, \nu \leftarrow \nu_{\mathrm{NE}}$ and dividing by T, we get

$$\frac{1}{T}\sum_{t=1}^{T} \Big(\langle \mu_t - \mu_{\mathrm{NE}}, -g + G\nu_t \rangle + \left\langle \nu_t - \nu_{\mathrm{NE}}, -G^\dagger \mu_t) \right\rangle \Big) \le \frac{D_0}{\eta T} + \frac{\eta M^2}{4}. \tag{D.32}$$

The left-hand side of (D.32) can be simplified to

$$\begin{aligned} \frac{1}{T}\sum_{t=1}^{T} \Big(\langle \mu_t - \mu_{\mathrm{NE}}, -g + G\nu_t \rangle + \left\langle \nu_t - \nu_{\mathrm{NE}}, -G^\dagger \mu_t \right\rangle \Big) &= \frac{1}{T}\sum_{t=1}^{T} \Big(\langle \mu_{\mathrm{NE}} - \mu_t, g \rangle - \langle \mu_{\mathrm{NE}}, G\nu_t \rangle + \langle \mu_t, G\nu_{\mathrm{NE}} \rangle \Big) \\ &= \langle \mu_{\mathrm{NE}}, g - G\bar{\nu}_T \rangle - \langle \bar{\mu}_T, g - G\nu_{\mathrm{NE}} \rangle. \end{aligned} \tag{D.33}$$

By definition of the Nash Equilibrium, we have

$$\begin{aligned} \langle \bar{\mu}_T, g - G\nu_{\mathrm{NE}} \rangle &\le \langle \mu_{\mathrm{NE}}, g - G\nu_{\mathrm{NE}} \rangle \le \langle \mu_{\mathrm{NE}}, g - G\bar{\nu}_T \rangle, \\ \langle \bar{\mu}_T, g - G\nu_{\mathrm{NE}} \rangle &\le \quad \langle \bar{\mu}_T, g - G\bar{\nu}_T \rangle \quad \le \langle \mu_{\mathrm{NE}}, g - G\bar{\nu}_T \rangle, \end{aligned} \tag{D.34}$$

which implies

$$\left| \langle \bar{\mu}_T, g - G\bar{\nu}_T \rangle - \langle \mu_{\mathrm{NE}}, g - G\nu_{\mathrm{NE}} \rangle \right| \le \langle \mu_{\mathrm{NE}}, g - G\bar{\nu}_T \rangle - \langle \bar{\mu}_T, g - G\nu_{\mathrm{NE}} \rangle. \tag{D.35}$$

Combining (D.45)-(D.48), we conclude that

$$\eta = \frac{2}{M}\sqrt{\frac{D_0}{T}} \quad \Rightarrow \quad \left| \langle \bar{\mu}_T, g - G\bar{\nu}_T \rangle - \langle \mu_{\mathrm{NE}}, g - G\nu_{\mathrm{NE}} \rangle \right| \le M\sqrt{\frac{D_0}{T}}.$$

Mirror descent, stochastic derivatives

We first write

$$\langle \mu_t - \mu, \eta(-\hat{g} + \hat{G}\nu_t) \rangle = \langle \mu_t - \mu, \eta(-g + G\nu_t) \rangle + \left\langle \mu_t - \mu, \eta\Big[-\hat{g} + \hat{G}\nu_t + g - G\nu_t \Big] \right\rangle.$$

Taking conditional expectation and using the bias estimate of stochastic derivatives, we conclude that

$$\begin{aligned} \mathbb{E}\langle \mu_t - \mu, \eta(-\hat{g} + \hat{G}\nu_t) \rangle &\le \langle \mu_t - \mu, \eta(-g + G\nu_t) \rangle + \|\mu_t - \mu\|_{\mathrm{TV}} \cdot \eta\tau \\ &\le \langle \mu_t - \mu, \eta(-g + G\nu_t) \rangle + 2\eta\tau. \end{aligned}$$

Therefore, using exactly the same argument leading to (D.30), we may obtain

$$\mathbb{E}\sum_{t=1}^{T}\left\langle \mu_t-\mu,-\hat{g}+\hat{G}v_t\right\rangle \leq \frac{\mathbb{E}D_\Phi(\mu,\mu_1)}{\eta}+\frac{\eta M'^2T}{8}+2\eta T\tau.$$

The rest is the same as with deterministic derivatives.

D.2.3 Proof of convergence rates for infinite-dimensional Mirror-Prox

We first need a technical lemma, which is Lemma 6.2 of [JN11] tailored to our infinite-dimensional setting. We give a slightly different proof.

Lemma D.1. *Given any* $\mu\in\mathcal{M}(\mathcal{Z})$ *and* $h,h'\in\mathcal{F}(\mathcal{Z})$, *let* $\mu=\mathrm{MD}_\eta\left(\tilde{\mu},h\right)$ *and* $\tilde{\mu}_+=\mathrm{MD}_\eta\left(\tilde{\mu},h'\right)$. *Let* Φ *be* α*-strongly convex (recall that* $\alpha=4$ *when* Φ *is the entropy). Then, for any* $\mu_\star\in\mathcal{M}(\mathcal{Z})$, *we have*

$$\left\langle \mu-\mu_\star,\eta h'\right\rangle \leq D_\Phi(\mu_\star,\tilde{\mu})-D_\Phi(\mu_\star,\tilde{\mu}_+)+\frac{\eta^2}{2\alpha}\left\|h-h'\right\|^2_{\mathbb{L}^\infty}-\frac{\alpha}{2}\left\|\mu-\tilde{\mu}\right\|^2_{\mathrm{TV}}. \tag{D.36}$$

Proof. Recall from (D.9) that entropy is α-strongly convex with respect to $\|\cdot\|_{\mathrm{TV}}$. We first write

$$\left\langle \mu-\mu_\star,\eta h'\right\rangle = \left\langle \tilde{\mu}_+-\mu_\star,\eta h'\right\rangle+\left\langle \mu-\tilde{\mu}_+,\eta h\right\rangle+\left\langle \mu-\tilde{\mu}_+,\eta(h'-h)\right\rangle. \tag{D.37}$$

For the first term, (D.14) and (D.15) implies

$$\begin{aligned}\left\langle \tilde{\mu}_+-\mu_\star,\eta h'\right\rangle &= \left\langle \tilde{\mu}_+-\mu_\star,\mathrm{d}\Phi(\tilde{\mu})-\mathrm{d}\Phi(\tilde{\mu}_+)\right\rangle\\ &= -D_\Phi(\tilde{\mu}_+,\tilde{\mu})-D_\Phi(\mu_\star,\tilde{\mu}_+)+D_\Phi(\mu_\star,\tilde{\mu}).\end{aligned} \tag{D.38}$$

Similarly, the second term of the right-hand side of (D.37) can be written as

$$\left\langle \mu-\tilde{\mu}_+,\eta h\right\rangle = -D_\Phi(\mu,\tilde{\mu})-D_\Phi(\tilde{\mu}_+,\mu)+D_\Phi(\tilde{\mu}_+,\tilde{\mu}). \tag{D.39}$$

Hölder's inequality for the third term gives

$$\begin{aligned}\left\langle \mu-\tilde{\mu}_+,\eta(h'-h)\right\rangle &\leq \left\|\mu-\tilde{\mu}_+\right\|_{\mathrm{TV}}\left\|\eta(h'-h)\right\|_{\mathbb{L}^\infty}\\ &\leq \frac{\alpha}{2}\left\|\mu-\tilde{\mu}_+\right\|^2_{\mathrm{TV}}+\frac{1}{2\alpha}\left\|\eta(h'-h)\right\|^2_{\mathbb{L}^\infty}.\end{aligned} \tag{D.40}$$

Finally, recall that Φ is α-strongly convex, and hence we have

$$-D_\Phi(\tilde{\mu}_+,\mu)\leq -\frac{\alpha}{2}\left\|\mu-\tilde{\mu}_+\right\|^2_{\mathrm{TV}},\quad -D_\Phi(\mu,\tilde{\mu})\leq -\frac{\alpha}{2}\left\|\mu-\tilde{\mu}\right\|^2_{\mathrm{TV}}. \tag{D.41}$$

The lemma follows by combining inequalities (D.38)-(D.41) in (D.37). ■

Mirror-Prox, deterministic derivatives

Let $\alpha = 4$, $\bar\mu_T := \frac{1}{T}\sum_{t=1}^T \mu_t$, and $\bar\nu_T := \frac{1}{T}\sum_{t=1}^T \nu_t$.

In Lemma D.1, substituting $\mu_\star \leftarrow \mu_{\mathrm{NE}}$, $\tilde\mu \leftarrow \tilde\mu_t$, $h \leftarrow -g + G\tilde\nu_t$ (so that $\mu = \mu_t$) and $h' \leftarrow -g + G\nu_t$ (so that $\tilde\mu_+ = \tilde\mu_{t+1}$), we get

$$\left\langle \mu_t - \mu_{\mathrm{NE}}, \eta(-g + G\nu_t)\right\rangle \le D_\Phi(\mu_{\mathrm{NE}}, \tilde\mu_t) - D_\Phi(\mu_{\mathrm{NE}}, \tilde\mu_{t+1}) + \frac{\eta^2}{2\alpha}\|G(\nu_t - \tilde\nu_t)\|^2_{\mathbb{L}^\infty} - \frac{\alpha}{2}\left\|\tilde\mu_t - \mu_t\right\|^2_{\mathrm{TV}}. \tag{D.42}$$

Similarly, we have

$$\left\langle \nu_t - \nu_{\mathrm{NE}}, -\eta G^\dagger \mu_t\right\rangle \le D_\Phi(\nu_{\mathrm{NE}}, \tilde\nu_t) - D_\Phi(\nu_{\mathrm{NE}}, \tilde\nu_{t+1}) + \frac{\eta^2}{2\alpha}\left\|G^\dagger(\mu_t - \tilde\mu_t)\right\|^2_{\mathbb{L}^\infty} - \frac{\alpha}{2}\|\tilde\nu_t - \nu_t\|^2_{\mathrm{TV}}. \tag{D.43}$$

Since $\|G(\nu_t - \tilde\nu_t)\|_{\mathbb{L}^\infty} \le L\cdot\|\nu_t - \tilde\nu_t\|_{\mathrm{TV}}$ and $\left\|G^\dagger(\mu_t - \tilde\mu_t)\right\|_{\mathbb{L}^\infty} \le L\cdot\left\|\mu_t - \tilde\mu_t\right\|_{\mathrm{TV}}$, summing up (D.42) and (D.43) yields

$$\begin{aligned}
\left\langle \mu_t - \mu_{\mathrm{NE}}, \eta(-g + G\nu_t)\right\rangle + \left\langle \nu_t - \nu_{\mathrm{NE}}, -\eta G^\dagger \mu_t\right\rangle &\le D_\Phi(\mu_{\mathrm{NE}}, \tilde\mu_t) - D_\Phi(\mu_{\mathrm{NE}}, \tilde\mu_{t+1}) + D_\Phi(\nu_{\mathrm{NE}}, \tilde\nu_t) - D_\Phi(\nu_{\mathrm{NE}}, \tilde\nu_{t+1}) \\
&\quad + \left(\frac{\eta^2 L^2}{2\alpha} - \frac{\alpha}{2}\right)\left(\left\|\tilde\mu_t - \mu_t\right\|^2_{\mathrm{TV}} + \|\tilde\nu_t - \nu_t\|^2_{\mathrm{TV}}\right) \\
&\le D_\Phi(\mu_{\mathrm{NE}}, \tilde\mu_t) - D_\Phi(\mu_{\mathrm{NE}}, \tilde\mu_{t+1}) + D_\Phi(\nu_{\mathrm{NE}}, \tilde\nu_t) - D_\Phi(\nu_{\mathrm{NE}}, \tilde\nu_{t+1})
\end{aligned}$$

if $\eta \le \frac{\alpha}{L} = \frac{4}{L}$. Summing up the last inequality over t and using $D_\Phi(\cdot,\cdot) \ge 0$, we obtain

$$\frac{1}{T}\sum_{t=1}^T \left(\left\langle \mu_t - \mu_{\mathrm{NE}}, \eta(-g + G\nu_t)\right\rangle + \left\langle \nu_t - \nu_{\mathrm{NE}}, -\eta G^\dagger \mu_t\right\rangle\right) \le \frac{D_\Phi(\mu_{\mathrm{NE}}, \tilde\mu_1) + D_\Phi(\nu_{\mathrm{NE}}, \tilde\nu_1)}{T} = \frac{D_0}{T}. \tag{D.44}$$

The left-hand side of (D.44) can be simplified to

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^T \left(\left\langle \mu_t - \mu_{\mathrm{NE}}, \eta(-g + G\nu_t)\right\rangle + \left\langle \nu_t - \nu_{\mathrm{NE}}, -\eta G^\dagger \mu_t\right\rangle\right) &= \frac{\eta}{T}\sum_{t=1}^T \left(\left\langle \mu_{\mathrm{NE}} - \mu_t, g\right\rangle - \left\langle \mu_{\mathrm{NE}}, G\nu_t\right\rangle + \left\langle \mu_t, G\nu_{\mathrm{NE}}\right\rangle\right) \\
&= \eta\left(\left\langle \mu_{\mathrm{NE}}, g - G\bar\nu_T\right\rangle - \left\langle \bar\mu_T, g - G\nu_{\mathrm{NE}}\right\rangle\right).
\end{aligned} \tag{D.45}$$

By definition of the $(\mu_{\mathrm{NE}}, \nu_{\mathrm{NE}})$, we have

$$\begin{aligned}
\left\langle \bar\mu_T, g - G\nu_{\mathrm{NE}}\right\rangle &\le \left\langle \mu_{\mathrm{NE}}, g - G\nu_{\mathrm{NE}}\right\rangle \le \left\langle \mu_{\mathrm{NE}}, g - G\bar\nu_T\right\rangle, \\
\left\langle \bar\mu_T, g - G\nu_{\mathrm{NE}}\right\rangle &\le \quad \left\langle \bar\mu_T, g - G\bar\nu_T\right\rangle \quad \le \left\langle \mu_{\mathrm{NE}}, g - G\bar\nu_T\right\rangle,
\end{aligned} \tag{D.46}$$

which implies

$$|\langle\bar{\mu}_T, g - G\bar{\nu}_T\rangle - \langle\mu_{\mathrm{NE}}, g - G\nu_{\mathrm{NE}}\rangle| \le \langle\mu_{\mathrm{NE}}, g - G\bar{\nu}_T\rangle - \langle\bar{\mu}_T, g - G\nu_{\mathrm{NE}}\rangle. \tag{D.47}$$

Combining (D.44)-(D.47), we conclude

$$\eta \le \frac{4}{L} \quad \Rightarrow \quad |\langle\bar{\mu}_T, g - G\bar{\nu}_T\rangle - \langle\mu_{\mathrm{NE}}, g - G\nu_{\mathrm{NE}}\rangle| \le \frac{D_0}{T\eta}.$$

Mirror-Prox, stochastic derivatives

Let $\alpha = 4$, $\bar{\mu}_T := \frac{1}{T}\sum_{t=1}^T \mu_t$, and $\bar{\nu}_T := \frac{1}{T}\sum_{t=1}^T \nu_t$. Set the step-size to $\eta = \min\left[\frac{\alpha}{\sqrt{3}L}, \sqrt{\frac{\alpha D_0}{6T\sigma^2}}\right]$.

In **Lemma D.1**, substituting $\mu_\star \leftarrow \mu_{\mathrm{NE}}$, $\tilde{\mu} \leftarrow \tilde{\mu}_t, h \leftarrow -\hat{g} + \hat{G}\tilde{\nu}_t$ (so that $\mu = \mu_t$), and $h' \leftarrow -\hat{g} + \hat{G}\nu_t$ (so that $\tilde{\mu}_+ = \tilde{\mu}_{t+1}$), we get

$$\langle\mu_t - \mu_{\mathrm{NE}}, \eta(-\hat{g} + \hat{G}\nu_t)\rangle \le D_\Phi(\mu_{\mathrm{NE}}, \tilde{\mu}_t) - D_\Phi(\mu_{\mathrm{NE}}, \tilde{\mu}_{t+1}) + \frac{\eta^2}{2\alpha}\left\|\hat{G}\nu_t - \hat{G}\tilde{\nu}_t\right\|_{\mathbb{L}^\infty}^2 - \frac{\alpha}{2}\left\|\tilde{\mu}_t - \mu_t\right\|_{\mathrm{TV}}^2. \tag{D.48}$$

Note that

$$\begin{aligned}\mathbb{E}\left\|\hat{G}\nu_t - \hat{G}\tilde{\nu}_t\right\|_{\mathbb{L}^\infty}^2 &\le 3\left(\mathbb{E}\left\|\hat{G}\nu_t - G\nu_t\right\|_{\mathbb{L}^\infty}^2 + \mathbb{E}\left\|G\nu_t - G\tilde{\nu}_t\right\|_{\mathbb{L}^\infty}^2 + \mathbb{E}\left\|G\tilde{\nu}_t - \hat{G}\tilde{\nu}_t\right\|_{\mathbb{L}^\infty}^2\right)\\ &\le 6\sigma^2 + 3L^2\mathbb{E}\left\|\nu_t - \tilde{\nu}_t\right\|_{\mathrm{TV}}^2.\end{aligned}$$

Therefore, taking expectation conditioned on the history for both sides of (D.48) and using the bias estimates of the stochastic derivatives, we get

$$\begin{aligned}\langle\mu_t - \mu_{\mathrm{NE}}, \eta(-g + G\nu_t)\rangle \le \mathbb{E}D_\Phi(\mu_{\mathrm{NE}}, \tilde{\mu}_t) - \mathbb{E}D_\Phi(\mu_{\mathrm{NE}}, \tilde{\mu}_{t+1}) + \frac{3\eta^2\sigma^2}{\alpha}&\\ + \frac{3\eta^2L^2}{2\alpha}\mathbb{E}\left\|\nu_t - \tilde{\nu}_t\right\|_{\mathrm{TV}}^2 - \frac{\alpha}{2}\mathbb{E}\left\|\tilde{\mu}_t - \mu_t\right\|_{\mathrm{TV}}^2 + 2\eta\tau.&\end{aligned}$$

Similarly, we have

$$\begin{aligned}\left\langle\nu_t - \nu_{\mathrm{NE}}, -\eta G^\dagger\mu_t\right\rangle \le \mathbb{E}D_\Phi(\nu_{\mathrm{NE}}, \tilde{\nu}_t) - \mathbb{E}D_\Phi(\nu_{\mathrm{NE}}, \tilde{\nu}_{t+1}) + \frac{3\eta^2\sigma^2}{\alpha}&\\ + \frac{3\eta^2L^2}{2\alpha}\mathbb{E}\left\|\mu_t - \tilde{\mu}_t\right\|_{\mathrm{TV}}^2 - \frac{\alpha}{2}\mathbb{E}\left\|\tilde{\nu}_t - \nu_t\right\|_{\mathrm{TV}}^2 + 2\eta\tau.&\end{aligned}$$

Summing up the last two inequalities over t with $\eta \leq \frac{\alpha}{\sqrt{3}L}$ then yields

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\Big(\langle\mu_t-\mu_{\text{NE}},-g+G\nu_t\rangle+\Big\langle\nu_t-\nu_{\text{NE}},-G^\dagger\mu_t)\Big\rangle\Big) &\leq \frac{D_0}{\eta T}+\frac{6\eta\sigma^2}{\alpha}+4\tau\\ &\leq \max\left[2\sqrt{\frac{6\sigma^2 D_0}{\alpha T}},\frac{2\sqrt{3}LD_0}{\alpha T}\right]+4\tau.\end{aligned}$$

by definition of η. The rest is the same as with deterministic derivatives.

Algorithm 10: Approx Inf Mirror Decent

```
Require: $W[1], \Theta[1] \leftarrow n'$ samples from random initialization, $\{\gamma_t\}_{t=1}^{T-1}, \{\epsilon_t\}_{t=1}^{T-1}, \{K\}_{t=1}^{T-1}, n, n'$,
  standard normal noise $\xi_k, \xi'_k$.
  for $t = 1, 2, \ldots, T-1$ do
    $C \leftarrow \cup_{s=1}^{t} W[s], \quad D \leftarrow \cup_{s=1}^{t} \Theta[s]$
    $\boldsymbol{w}_t^{(1)} \leftarrow \mathrm{UNIF}(W[t]), \quad \boldsymbol{\theta}_t^{(1)} \leftarrow \mathrm{UNIF}(\Theta[t])$
    for $k = 1, 2, \ldots, K_t, \ldots, K_t + n'$ do
      Generate $A = \{X_1, \ldots, X_n\} \sim \mathbb{P}_{\boldsymbol{\theta}_t^{(k)}}$
      $\boldsymbol{\theta}_t^{(k+1)} = \boldsymbol{\theta}_t^{(k)} + \frac{\gamma_t}{nn'} \nabla_{\boldsymbol{\theta}} \sum_{X_i \in A} \sum_{\boldsymbol{w} \in C} f_{\boldsymbol{w}}(X_i) + \sqrt{2\gamma_t} \epsilon_t \xi_k$
      Generate $B = \{X_1^{\mathrm{real}}, \ldots, X_n^{\mathrm{real}}\} \sim \mathbb{P}_{\mathrm{real}}$
      $B' \leftarrow \{\}$
      for each $\boldsymbol{\theta} \in D$ do
        Generate $\tilde{B} = \{X'_1, \ldots, X'_n\} \sim \mathbb{P}_{\boldsymbol{\theta}}$
        $B' \leftarrow B' \cup \tilde{B}$
      end for
      $$\boldsymbol{w}_t^{(k+1)} = \boldsymbol{w}_t^{(k)} + \frac{\gamma_t t}{n} \nabla_{\boldsymbol{w}} \sum_{X_i^{\mathrm{real}} \in B} f_{\boldsymbol{w}_t^{(k)}}(X_i^{\mathrm{real}}) - \frac{\gamma_t}{nn'} \nabla_{\boldsymbol{w}} \sum_{X'_i \in B'} f_{\boldsymbol{w}_t^{(k)}}(X'_i) + \sqrt{2\gamma_t} \epsilon_t \xi'_k$$
    end for
    $W[t+1] \leftarrow \left\{\boldsymbol{w}_t^{(K+1)}, \ldots, \boldsymbol{w}_t^{(K+n')}\right\}, \quad \Theta[t+1] \leftarrow \left\{\boldsymbol{\theta}_t^{(K+1)}, \ldots, \boldsymbol{\theta}_t^{(K+n')}\right\}$
  end for
  idx ← $\mathrm{UNIF}(1, 2, \ldots, T)$
return $W$[idx], $\Theta$[idx].
```

D.3 Omitted Pseudocodes in the Main Text

We use the following notation for the hyperparameters of our algorithms:

n : number of samples in the data batch.

n' : number of samples for each probability measure.

γ_t : SGLD step-size at iteration t.

ϵ_t : thermal noise of SGLD at iteration t.

K_t : warmup steps for SGLD at iteration t.

β : exponential damping factor in the weighted average.

The approximate infinite-dimensional entropic MD and MP in Section 5.4.1 are depicted in Algorithms 10–11, respectively. Algorithm 12 gives the heuristic version of the entropic Mirror-Prox.

Algorithm 11: Approx Inf Mirror-Prox

```
Require: $\tilde{W}[1], \tilde{\Theta}[1] \leftarrow n'$ samples from random initialization, $\{\gamma_t\}_{t=1}^{T}, \{\epsilon_t\}_{t=1}^{T}, \{K_t\}_{t=1}^{T}, n, n'$,
  standard normal noise $\xi_k, \xi'_k, \xi''_k, \xi'''_k$.
  for $t = 1, 2, \ldots, T$ do
    $C \leftarrow \tilde{W}[t] \cup \left(\cup_{s=1}^{t-1} W[s]\right), \quad D \leftarrow \tilde{\Theta}[t] \cup \left(\cup_{s=1}^{t-1} \Theta[s]\right)$
    $\boldsymbol{w}_t^{(1)} \leftarrow \mathrm{UNIF}(\tilde{W}[t]), \quad \boldsymbol{\theta}_t^{(1)} \leftarrow \mathrm{UNIF}(\tilde{\Theta}[t])$
    for $k = 1, 2, \ldots, K_t, \ldots, K_t + n'$ do
      Generate $A = \{X_1, \ldots, X_n\} \sim \mathbb{P}_{\boldsymbol{\theta}_t^{(k)}}$
      $\boldsymbol{\theta}_t^{(k+1)} = \boldsymbol{\theta}_t^{(k)} + \frac{\gamma_t}{nn'} \nabla_{\boldsymbol{\theta}} \sum_{X_i \in A} \sum_{\boldsymbol{w} \in C} f_{\boldsymbol{w}}(X_i) + \sqrt{2\gamma_t} \epsilon_t \xi_k$
      Generate $B = \{X_1^{\mathrm{real}}, \ldots, X_n^{\mathrm{real}}\} \sim \mathbb{P}_{\mathrm{real}}$
      $B' \leftarrow \{\}$
      for each $\boldsymbol{\theta} \in D$ do
        Generate $\tilde{B} = \{X'_1, \ldots, X'_n\} \sim \mathbb{P}_{\boldsymbol{\theta}}$
        $B' \leftarrow B' \cup \tilde{B}$
      end for
      $$\boldsymbol{w}_t^{(k+1)} = \boldsymbol{w}_t^{(k)} + \frac{\gamma_t t}{n} \nabla_{\boldsymbol{w}} \sum_{X_i^{\mathrm{real}} \in B} f_{\boldsymbol{w}_t^{(k)}}(X_i^{\mathrm{real}}) - \frac{\gamma_t}{nn'} \nabla_{\boldsymbol{w}} \sum_{X'_i \in B'} f_{\boldsymbol{w}_t^{(k)}}(X'_i) + \sqrt{2\gamma_t} \epsilon_t \xi'_k$$
    end for
    $W[t] \leftarrow \left\{\boldsymbol{w}_t^{(K+1)}, \ldots, \boldsymbol{w}_t^{(K+n')}\right\}, \quad \Theta[t] \leftarrow \left\{\boldsymbol{\theta}_t^{(K+1)}, \ldots, \boldsymbol{\theta}_t^{(K+n')}\right\}$
    $C' \leftarrow \cup_{s=1}^{t} W[s], \quad D' \leftarrow \cup_{s=1}^{t} \Theta[s]$
    $\tilde{\boldsymbol{w}}_{t+1}^{(1)} \leftarrow \mathrm{UNIF}(\tilde{W}[t]), \quad \tilde{\boldsymbol{\theta}}_{t+1}^{(1)} \leftarrow \mathrm{UNIF}(\tilde{\Theta}[t])$
    for $k = 1, 2, \ldots, K_t, \ldots, K_t + n'$ do
      Generate $A = \{X_1, \ldots, X_n\} \sim \mathbb{P}_{\tilde{\boldsymbol{\theta}}_t^{(k)}}$
      $\tilde{\boldsymbol{\theta}}_{t+1}^{(k+1)} = \tilde{\boldsymbol{\theta}}_{t+1}^{(k)} + \frac{\gamma_t}{nn'} \nabla_{\boldsymbol{\theta}} \sum_{X_i \in A} \sum_{\boldsymbol{w} \in C'} f_{\boldsymbol{w}}(X_i) + \sqrt{2\gamma_t} \epsilon_t \xi''_k$
      Generate $B = \{X_1^{\mathrm{real}}, \ldots, X_n^{\mathrm{real}}\} \sim \mathbb{P}_{\mathrm{real}}$
      $B' \leftarrow \{\}$
      for each $\boldsymbol{\theta} \in D'$ do
        Generate $\tilde{B} = \{X'_1, \ldots, X'_n\} \sim \mathbb{P}_{\boldsymbol{\theta}}$
        $B' \leftarrow B' \cup \tilde{B}$
      end for
      $$\tilde{\boldsymbol{w}}_{t+1}^{(k+1)} = \tilde{\boldsymbol{w}}_{t+1}^{(k)} + \frac{\gamma_t t}{n} \nabla_{\boldsymbol{w}} \sum_{X_i^{\mathrm{real}} \in B} f_{\tilde{\boldsymbol{w}}_{t+1}^{(k)}}(X_i^{\mathrm{real}}) - \frac{\gamma_t}{nn'} \nabla_{\boldsymbol{w}} \sum_{X'_i \in B'} f_{\tilde{\boldsymbol{w}}_{t+1}^{(k)}}(X'_i) + \sqrt{2\gamma_t} \epsilon_t \xi'''_k)$$
    end for
    $\tilde{W}[t+1] \leftarrow \left\{\tilde{\boldsymbol{w}}_{t+1}^{(K+1)}, \ldots, \tilde{\boldsymbol{w}}_{t+1}^{(K+n')}\right\}, \quad \tilde{\Theta}[t+1] \leftarrow \left\{\tilde{\boldsymbol{\theta}}_{t+1}^{(K+1)}, \ldots, \tilde{\boldsymbol{\theta}}_{t+1}^{(K+n')}\right\}$
  end for
  idx $\leftarrow \mathrm{UNIF}(1, 2, \ldots, T)$
return $W$[idx], $\Theta$[idx].
```

Algorithm 12: Mirror-Prox-GAN: Approximate Mirror-Prox for GANs

```
Require: $\tilde{\boldsymbol{w}}_1, \tilde{\boldsymbol{\theta}}_1 \leftarrow$ random initialization, $\boldsymbol{w}_0 \leftarrow \tilde{\boldsymbol{w}}_1, \boldsymbol{\theta}_0 \leftarrow \tilde{\boldsymbol{\theta}}_1, \{\gamma_t\}_{t=1}^T, \{\epsilon_t\}_{t=1}^T, \{K_t\}_{t=1}^T, \beta$, standard normal noise $\xi_k, \xi'_k, \xi''_k, \xi'''_k$.
  for $t = 1, 2, \ldots, T$ do
    $\bar{\boldsymbol{w}}_t, \bar{\boldsymbol{w}}_{t+1}, \tilde{\boldsymbol{w}}_t^{(1)}, \tilde{\boldsymbol{w}}_{t+1}^{(1)} \leftarrow \tilde{\boldsymbol{w}}_t, \quad \bar{\boldsymbol{\theta}}_t, \bar{\boldsymbol{\theta}}_{t+1}, \tilde{\boldsymbol{\theta}}_t^{(1)}, \tilde{\boldsymbol{\theta}}_{t+1}^{(1)} \leftarrow \tilde{\boldsymbol{\theta}}_t$
    for $k = 1, 2, \ldots, K_t$ do
      Generate $A = \{X_1, \ldots, X_n\} \sim \mathbb{P}_{\boldsymbol{\theta}_t^{(k)}}$
      $\boldsymbol{\theta}_t^{(k+1)} = \boldsymbol{\theta}_t^{(k)} + \frac{\gamma_t}{n} \nabla_{\boldsymbol{\theta}} \sum_{X_i \in A} f_{\tilde{\boldsymbol{w}}_t}(X_i) + \sqrt{2\gamma_t} \epsilon_t \xi_k$
      Generate $B = \{X_1^{\mathrm{real}}, \ldots, X_n^{\mathrm{real}}\} \sim \mathbb{P}_{\mathrm{real}}$
      Generate $B' = \{X'_1, \ldots, X'_n\} \sim \mathbb{P}_{\tilde{\boldsymbol{\theta}}_t}$
      $\boldsymbol{w}_t^{(k+1)} = \boldsymbol{w}_t^{(k)} + \frac{\gamma_t}{n} \nabla_{\boldsymbol{w}} \sum_{X_i^{\mathrm{real}} \in B} f_{\boldsymbol{w}_t^{(k)}}(X_i^{\mathrm{real}}) - \frac{\gamma_t}{n} \nabla_{\boldsymbol{w}} \sum_{X'_i \in B'} f_{\boldsymbol{w}_t^{(k)}}(X'_i) + \sqrt{2\gamma_t} \epsilon_t \xi'_k$
      $\bar{\boldsymbol{w}}_t \leftarrow (1-\beta) \bar{\boldsymbol{w}}_t + \beta \boldsymbol{w}_t^{(k+1)}$
      $\bar{\boldsymbol{\theta}}_t \leftarrow (1-\beta) \bar{\boldsymbol{\theta}}_t + \beta \boldsymbol{\theta}_t^{(k+1)}$
    end for
    $\boldsymbol{w}_t \leftarrow (1-\beta) \boldsymbol{w}_{t-1} + \beta \bar{\boldsymbol{w}}_t$
    $\boldsymbol{\theta}_t \leftarrow (1-\beta) \boldsymbol{\theta}_{t-1} + \beta \bar{\boldsymbol{\theta}}_t$

    for $k = 1, 2, \ldots, K_t$ do
      Generate $A = \{X_1, \ldots, X_n\} \sim \mathbb{P}_{\tilde{\boldsymbol{\theta}}_{t+1}^{(k)}}$
      $\tilde{\boldsymbol{\theta}}_{t+1}^{(k+1)} = \tilde{\boldsymbol{\theta}}_{t+1}^{(k)} + \frac{\gamma_t}{n} \nabla_{\boldsymbol{\theta}} \sum_{X_i \in A} f_{\boldsymbol{w}_t}(X_i) + \sqrt{2\gamma_t} \epsilon_t \xi''_k$
      Generate $B = \{X_1^{\mathrm{real}}, \ldots, X_n^{\mathrm{real}}\} \sim \mathbb{P}_{\mathrm{real}}$
      Generate $B' = \{X'_1, \ldots, X'_n\} \sim \mathbb{P}_{\boldsymbol{\theta}_t}$

      $\boldsymbol{w}_{t+1}^{(k+1)} = \boldsymbol{w}_{t+1}^{(k)} + \frac{\gamma_t}{n} \nabla_{\boldsymbol{w}} \sum_{X_i^{\mathrm{real}} \in B} f_{\boldsymbol{w}_{t+1}^{(k)}}(X_i^{\mathrm{real}}) - \frac{\gamma_t}{n} \nabla_{\boldsymbol{w}} \sum_{X'_i \in B'} f_{\boldsymbol{w}_{t+1}^{(k)}}(X'_i) + \sqrt{2\gamma_t} \epsilon_t \xi'''_k$

      $\bar{\boldsymbol{w}}_{t+1} \leftarrow (1-\beta) \bar{\boldsymbol{w}}_{t+1} + \beta \boldsymbol{w}_{t+1}^{(k+1)}$
      $\bar{\boldsymbol{\theta}}_{t+1} \leftarrow (1-\beta) \bar{\boldsymbol{\theta}}_{t+1} + \beta \boldsymbol{\theta}_{t+1}^{(k+1)}$
    end for
    $\tilde{\boldsymbol{w}}_{t+1} \leftarrow (1-\beta) \tilde{\boldsymbol{w}}_t + \beta \bar{\boldsymbol{w}}_{t+1}$
    $\tilde{\boldsymbol{\theta}}_{t+1} \leftarrow (1-\beta) \tilde{\boldsymbol{\theta}}_t + \beta \bar{\boldsymbol{\theta}}_{t+1}$
  end for
return $\boldsymbol{w}_T, \boldsymbol{\theta}_T$.
```

D.4 Algorithms and omitted proofs for Section 5.5

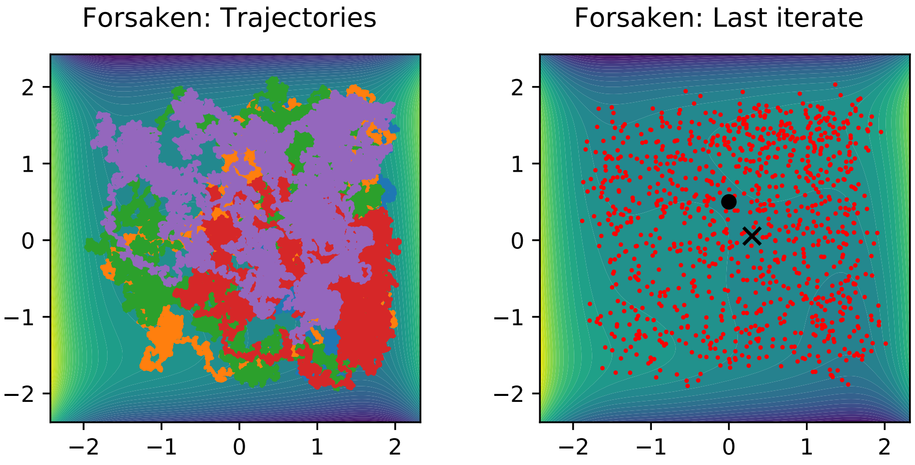
D.4.1 Algorithms and hyperparameters

The pseudocode of the algorithms can be found in Algorithm 13 (the symbol Π denotes the projection). The hyperparameter setting for experiments in Section 5.5 is:

- Algorithm 13 with GDA, and $\eta_t = 0.1$
- Algorithm 13 with EG, and $\eta_t = 0.1$
- Algorithm 13 with MixedNE-LD, $\eta_t = 0.1$, $\epsilon_t = 0.01$, $K_t = 50$, and $\beta = 0.5$.

We also note that we focus on the "last iterate" convergence for EG [ALW19, DP19], instead of the usual ergodic average in convex optimization literature. This is because, in practice, people almost exclusively use the last iterate.

D.4.2 MixedNE-LD on Example 4.5.2



(a) Different runs of MixedNE-LD. (b) Last iterates of MixedNE-LD.

Figs. D.1(a)–D.1(b) illustrate our MixedNE-LD applied to the objective in Example 4.5.2, on which most existing methods *provably* fail, as established in Section 4.5. In contrast, as can be inferred from Fig. D.1(b), MixedNE-LD can sometimes approach the desirable solutions.

D.4.3 Proof of Theorem 5.3

We will focus on the case $F(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}^2\boldsymbol{y}^2 - \boldsymbol{x}\boldsymbol{y}$. Without loss of generality, we may also assume that $\boldsymbol{y}(0) > \boldsymbol{x}(0) > 0$; the proof of the other cases follows the same argument.

Let $(\boldsymbol{x}(t),\boldsymbol{y}(t))$ follow the dynamics (5.9) with $\boldsymbol{x}(0)\cdot\boldsymbol{y}(0) > 0.5$. Assume, for the moment, that both $\boldsymbol{x}$ and $\boldsymbol{y}$ are without constraint. Then we have

$$\begin{aligned}\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{x}(t)^2+\boldsymbol{y}(t)^2\right) &= \boldsymbol{x}\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}+\boldsymbol{y}\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}\\ &= 2\boldsymbol{x}^2\boldsymbol{y}^2-\boldsymbol{x}\boldsymbol{y}+(-2\boldsymbol{x}^2\boldsymbol{y}^2+\boldsymbol{x}\boldsymbol{y})\\ &= 0\end{aligned}$$

implying that $\boldsymbol{x}^2(t)+\boldsymbol{y}^2(t) = \boldsymbol{x}^2(0)+\boldsymbol{y}^2(0)$ for all t. Therefore $\left(r\cos\left(t+\phi_1\right), r\sin\left(t+\phi_2\right)\right)$, where $(r\cos\phi_1, r\sin\phi_2) = (\boldsymbol{x}(0),\boldsymbol{y}(0))$, is a solution for dynamics for small enough t.

On the other hand, we have

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{x}(t)\boldsymbol{y}(t)\right) &= \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}(t)\cdot\boldsymbol{y}(t)+\boldsymbol{x}(t)\cdot\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}(t)\\ &= 2\boldsymbol{x}(t)\boldsymbol{y}^3(t)-\boldsymbol{y}^2(t)+\left(-2\boldsymbol{x}^3(t)\boldsymbol{y}(t)+\boldsymbol{x}^2(t)\right)\\ &= \left(\boldsymbol{x}^2(t)-\boldsymbol{y}^2(t)\right)\left(1-2\boldsymbol{x}(t)\boldsymbol{y}(t)\right)\\ &= \left(\boldsymbol{x}^2(t)-\boldsymbol{y}^2(t)\right)\left(1-2r^2\cos\left(t+\phi_1\right)\sin\left(t+\phi_2\right)\right).\end{aligned}$$

When $t=0$, we have $1-2r^2\cos\left(t+\phi_1\right)\sin\left(t+\phi_2\right)=1-2\boldsymbol{x}(0)\boldsymbol{y}(0)<0$. When $t=\frac{\pi}{t}$, we have

$$\begin{aligned}2r^2\cos\left(t+\phi_1\right)\sin\left(t+\phi_2\right) &= 2r^2\left(\frac{\sqrt{2}}{2}\cos\phi_1-\frac{\sqrt{2}}{2}\sin\phi_1\right)\left(\frac{\sqrt{2}}{2}\cos\phi_2+\frac{\sqrt{2}}{2}\sin\phi_2\right)\\ &= \left(\boldsymbol{x}(0)-\sqrt{r^2-\boldsymbol{x}(0)^2}\right)\left(\sqrt{r^2-\boldsymbol{y}(0)^2}+\boldsymbol{y}(0)\right)\\ &= (\boldsymbol{x}^2(0)-\boldsymbol{y}^2(0))<0\end{aligned}$$

whence $1-2r^2\cos\left(t+\phi_1\right)\sin\left(t+\phi_2\right)>0$. The intermediate value theorem then implies that there exists a $\tilde{t}$ such that $1-2\boldsymbol{x}(\tilde{t})\boldsymbol{y}(\tilde{t})=0$. But since $\{(\boldsymbol{x},\boldsymbol{y})\mid 2\boldsymbol{x}\boldsymbol{y}=1\}$ are the stationary points of the dynamics (5.9), we conclude that $\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{x}(t)\boldsymbol{y}(t)\right)=0$ whenever $t\geq\tilde{t}$; that is, $(\boldsymbol{x}(t),\boldsymbol{y}(t))$ gets trapped at the stationary point $(\boldsymbol{x}(\tilde{t}),\boldsymbol{y}(\tilde{t}))$. The concludes the first part the theorem when there is no boundary.

If the boundary is present, the dynamics (5.9) should be modified to the *projected dynamics* [BCL⁺18] and the proof remains the same, except that when $(\boldsymbol{x}(t),\boldsymbol{y}(t))$ hits the boundary, the curve needs to traverse along the boundary to decrease the norm.

We now turn to the statement for MixedNE-LD. Let $(\boldsymbol{x}_1,\boldsymbol{y}_1)$ be initialized at any stationary point: $\boldsymbol{x}_1\boldsymbol{y}_1=0.5$. Consider the two-step evolution of MixedNE-LD:

$$\begin{aligned}\boldsymbol{x}_2 &= \boldsymbol{x}_1+\sqrt{2\eta}\xi,\\ \boldsymbol{y}_2 &= \boldsymbol{y}_1+\sqrt{2\eta}\xi',\\ \boldsymbol{x}_3 &= \boldsymbol{x}_2+\eta\left(2\boldsymbol{x}_2\boldsymbol{y}_2^2-\boldsymbol{y}_2\right)+\sqrt{2\eta}\xi'',\\ \boldsymbol{y}_3 &= \boldsymbol{y}_2-\eta\left(2\boldsymbol{x}_2^2\boldsymbol{y}_2-\boldsymbol{x}_2\right)+\sqrt{2\eta}\xi'''\end{aligned}$$

where ξ,ξ',ξ'', and ξ''' are independent standard Gaussian. Since we initialize at a stationary point $\boldsymbol{x}_1\boldsymbol{y}_1=0.5$, we have

$$\begin{aligned}2\boldsymbol{x}_2\boldsymbol{y}_2-1 &= 2\boldsymbol{x}_1\boldsymbol{y}_1+\sqrt{2\eta}\boldsymbol{y}_1\xi+\sqrt{2\eta}\boldsymbol{x}_1\xi'+2\eta\xi\xi'-1\\ &= \sqrt{2\eta}\boldsymbol{y}_1\xi+\sqrt{2\eta}\boldsymbol{x}_1\xi'+2\eta\xi\xi'.\end{aligned}\tag{D.49}$$

Using the towering property of the expectation, (D.49), and the fact that ξ,ξ',ξ'', and ξ''' are independent standard Gaussian, we compute

$$\begin{aligned}
\mathbb{E}\boldsymbol{x}_3\boldsymbol{y}_3 &= \mathbb{E}\left[\mathbb{E}\left[\boldsymbol{x}_3\boldsymbol{y}_3 \mid \boldsymbol{x}_2, \boldsymbol{y}_2\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\left(\boldsymbol{x}_2 + \eta\left(2\boldsymbol{x}_2\boldsymbol{y}_2^2 - \boldsymbol{y}_2\right) + \sqrt{2\eta}\xi''\right)\left(\boldsymbol{y}_2 - \eta\left(2\boldsymbol{x}_2^2\boldsymbol{y}_2 - \boldsymbol{x}_2\right) + \sqrt{2\eta}\xi'''\right) \mid \boldsymbol{x}_2, \boldsymbol{y}_2\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\left(\boldsymbol{x}_2 + \eta\left(2\boldsymbol{x}_2\boldsymbol{y}_2^2 - \boldsymbol{y}_2\right)\right)\left(\boldsymbol{y}_2 - \eta\left(2\boldsymbol{x}_2^2\boldsymbol{y}_2 - \boldsymbol{x}_2\right)\right) \mid \boldsymbol{x}_2, \boldsymbol{y}_2\right]\right] \\
&= \mathbb{E}\left[\left(\boldsymbol{x}_2 + \eta\boldsymbol{y}_2\left(2\boldsymbol{x}_2\boldsymbol{y}_2 - 1\right)\right)\left(\boldsymbol{y}_2 - \eta\boldsymbol{x}_2\left(2\boldsymbol{x}_2\boldsymbol{y}_2 - 1\right)\right)\right] \\
&= \mathbb{E}\left[\boldsymbol{x}_2\boldsymbol{y}_2 - \eta\boldsymbol{x}_2^2\left(2\boldsymbol{x}_2\boldsymbol{y}_2 - 1\right) + \eta\boldsymbol{y}_2^2\left(2\boldsymbol{x}_2\boldsymbol{y}_2 - 1\right) - \eta^2\boldsymbol{x}_2\boldsymbol{y}_2\left(2\boldsymbol{x}_2\boldsymbol{y}_2 - 1\right)^2\right] \\
&= \mathbb{E}\Big[\boldsymbol{x}_1\boldsymbol{y}_1 - \eta\left(\boldsymbol{x}_1^2 + 2\eta\xi^2 + 2\sqrt{2\eta}\boldsymbol{x}_1\xi - \boldsymbol{y}_1^2 - 2\eta\xi'^2 - 2\sqrt{2\eta}\boldsymbol{y}_1\xi'\right)\left(\sqrt{2\eta}\boldsymbol{y}_1\xi + \sqrt{2\eta}\boldsymbol{x}_1\xi' + 2\eta\xi\xi'\right) \\
&\quad - 4\eta^2\left(\sqrt{2\eta}\boldsymbol{y}_1\xi + \sqrt{2\eta}\boldsymbol{x}_1\xi' + 2\eta\xi\xi'\right)\Big(2\eta\boldsymbol{y}_1^2\xi^2 + 2\eta\boldsymbol{x}_1^2\xi'^2 + 4\eta^2\xi^2\xi'^2 + 2\eta\xi\xi' + 4\sqrt{2}\eta^{\frac{3}{2}}\boldsymbol{x}_1\xi\xi'^2 \\
&\quad + 4\sqrt{2}\eta^{\frac{3}{2}}\boldsymbol{y}_2\xi^2\xi'\Big)\Big] \\
&= \boldsymbol{x}_1\boldsymbol{y}_1 - 0 - 4\eta^2\left(\eta\boldsymbol{y}_2^2 + \eta\boldsymbol{x}_1^2 + 2\eta^2 + 4\eta^2 + 4\eta^2 + 4\eta^2\right) \\
&= \boldsymbol{x}_1\boldsymbol{y}_1 - 4\eta^2\left(\eta\left(\boldsymbol{x}_1^2 + \boldsymbol{y}_1^2\right) + 14\eta^2\right)
\end{aligned}$$

which is (5.12).

D.4.4 Proof of Theorem 5.4

Spelling out the Newton dynamics (5.13), we get

$$\begin{aligned}
\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}(t) &= \frac{1}{2\boldsymbol{y}^2(t)}\left(2\boldsymbol{x}(t)\boldsymbol{y}^2(t) - \boldsymbol{y}(t)\right) \\
&= \boldsymbol{x}(t) - \frac{1}{2\boldsymbol{y}(t)}
\end{aligned}$$

and similarly $\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}(t) = -\boldsymbol{y}(t) + \frac{1}{2\boldsymbol{x}(t)}$. As a result, we have

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{x}(t)\boldsymbol{y}(t)\right) &= \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}(t)\cdot\boldsymbol{y}(t) + \boldsymbol{x}(t)\cdot\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}(t) \\
&= \boldsymbol{x}(t)\boldsymbol{y}(t) - \frac{1}{2} - \boldsymbol{x}(t)\boldsymbol{y}(t) + \frac{1}{2} \\
&= 0
\end{aligned}$$

which concludes the proof.

D.5 Details and further results for Section 5.6

This section contains all the details regarding our experiments in Section 5.6, as well as more results on synthetic and real datasets.

Network Architectures: For all experiments, we consider the gradient-penalized discriminator

Algorithm	**SGD**		**RMSProp**	**Adam**			**Entropic MD/MP**		
Dataset	S	M	L	S	M	L	S	M	L
Step-size γ	10^{-2}		10^{-4}	10^{-4}			10^{-2}		10^{-4}
Gradient penalty λ	0.1	10		0.1	10		0.1	10	
Noise ϵ							10^{-2}	10^{-3}	10^{-6}
Batch Size n	1024	50	64	1024	50	64	1024	50	64

Table D.1: Hyperparameter setting. “S”, “M”, “L” stands for synthetic data, `MNIST` and `LSUN bedroom`, respectively. MD for `LSUN bedroom` uses a RMSProp preconditioner, so the step-size is the same as one in RMSProp.

[GAA$^+$17] as a soft constraint alternative to the original Wasserstein GANs, as it is known to achieve much better performance. The gradient penalty parameter is denoted by λ below.

For synthetic data, we use fully connected networks for both the generator and discriminator. They consist of three layers, each of them containing 512 neurons, with ReLU as nonlinearity.

For `MNIST`, we use convolutional neural networks identical to [GAA$^+$17] as the generator and discriminator.[1] The generator uses a sigmoid function to map the output to range $[0,1]$.

For `LSUN bedroom`, we use DCGAN [RMC15], except that the number of the channels in each layer is half of the original model, and the last sigmoid function of the discriminator is removed. The output of the generator is mapped to $[0,1]$ by hyperbolic tangent and a linear transformation. The architecture contains batch normalization layer to ensure the stability of the training. For our MixedNE-LD and Mirror-Prox-GAN, the Gaussian noise from SGLD is not added to parameters in batch normalization layers, as the batch normalization creates strong dependence among entries of the weight matrix and was not covered by our theory.

Hyperparameter setting: The hyperparameter setting is summarized in Table D.1. For baselines (SGD, RMSProp, Adam), we use the settings identical to [GAA$^+$17]. For our proposed MixedNE-LD and Mirror-Prox-GAN, we set the damping factor β to be 0.9. For K_t, γ_t and ϵ_t, we use the simple exponential scheduling:

$$
\begin{aligned}
K_t &= \lfloor (1+10^{-5})^t \rfloor. \\
\gamma_t &= \gamma \times (1-10^{-5})^t, && \gamma \text{ in Table D.1.} \\
\epsilon_t &= \epsilon \times (1-5\times 10^{-5})^t, && \epsilon \text{ in Table D.1.}
\end{aligned}
$$

The idea is that the initial iterations are very noisy, and hence it makes sense to take less SGLD steps. As the iteration counts grow, the algorithms learn more meaningful parameters, and we should increase the number of SGLD steps as well as decreasing the step-size γ_t and thermal noise ϵ_t to make the sampling more accurate. This is akin to the warmup steps in the sampling literature.

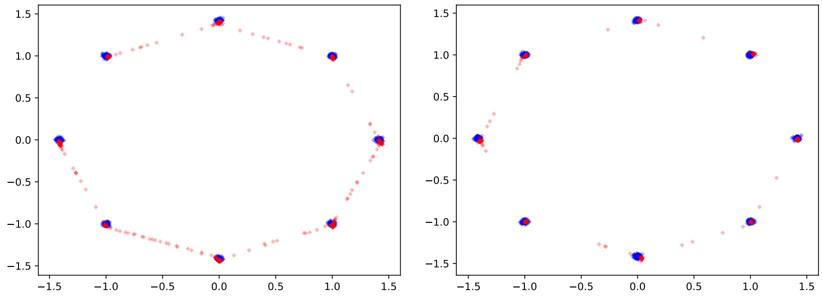
---

[1] Their code is available on https://github.com/igul222/improved_wgan_training.

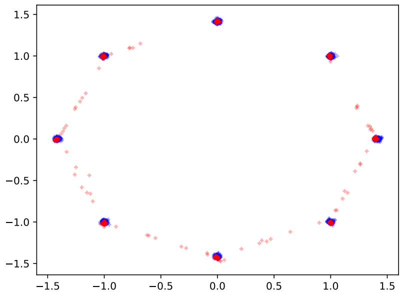
D.5.1 Synthetic Data

Figs. D.1–D.3 show results on learning 8 Gaussian mixtures, 25 Gaussian mixtures, and the Swiss Roll. As in the case for 25 Gaussian mixtures, we find that MixedNE-LD and Mirror-Prox-GAN can better capture the variance of the true distribution, as well as finding the unbiased modes.

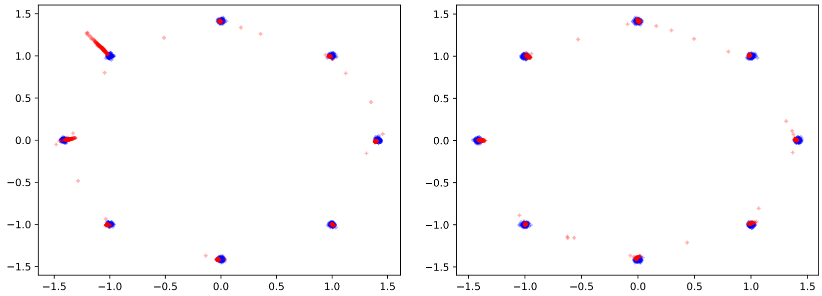
In Fig. D.4, we plot the data generated after $10^4, 2 \times 10^4, 5 \times 10^4, 8 \times 10^4$, and 10^5 iterations by different algorithms fro 25 Gaussian mixtures. It is clear that MixedNE-LD and Mirror-Prox-GAN find the modes of the distribution faster. In practice, it was observed that the noise introduced by SGLD quickly drives the iterates to non-trivial parameter regions, whereas SGD tends to get stuck at very bad local minima. Adam, as an adaptive algorithm, is capable of escaping bad local minima, however at a rate slower than MixedNE-LD and Mirror-Prox-GAN. The quality of Adam-based algorithms' final solutions are also not as good as MixedNE-LD and Mirror-Prox-GAN; see the discussions in Section 5.6.1.



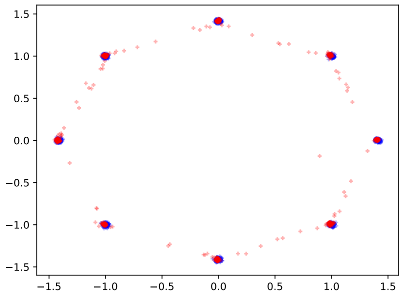
(c) SGD (d) Optimistic Adam



(e) MixedNE-LD

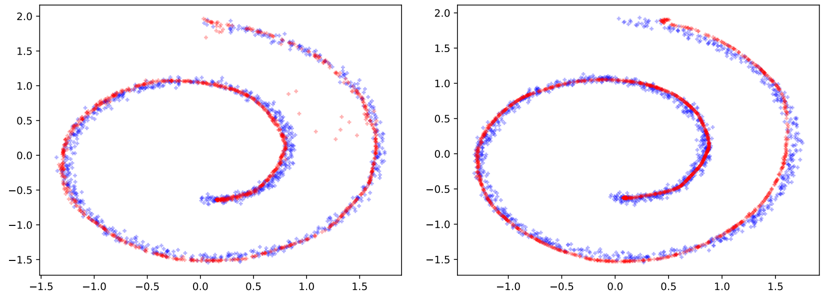


(f) Adam (g) Extra Adam

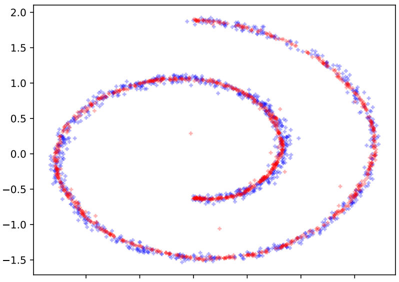


(h) Mirror-Prox-GAN

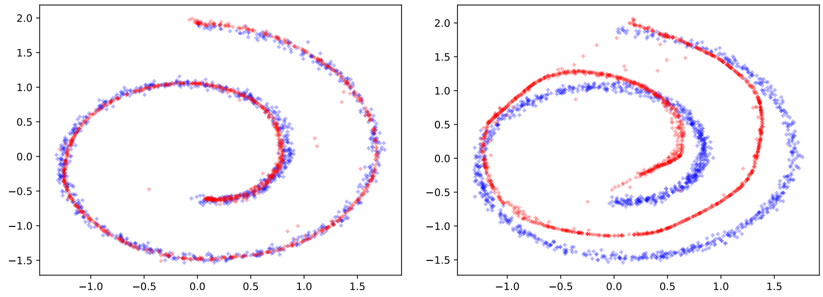
Figure D.1: Fitting 8 Gaussian mixtures up to 10^5 iterations.



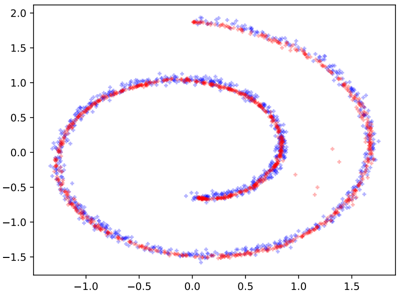
(a) SGD (b) Optimistic Adam



(c) MixedNE-LD

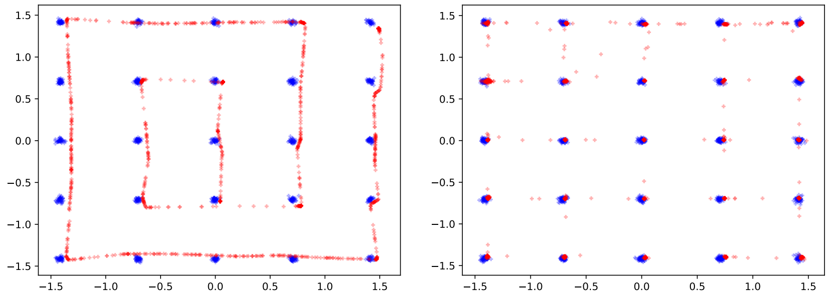


(d) Adam (e) Extra Adam

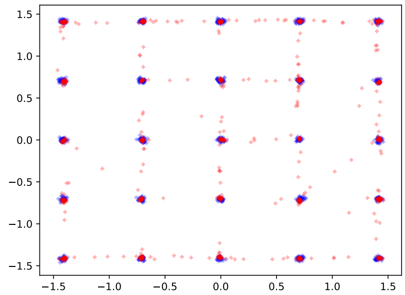


(f) Mirror-Prox-GAN

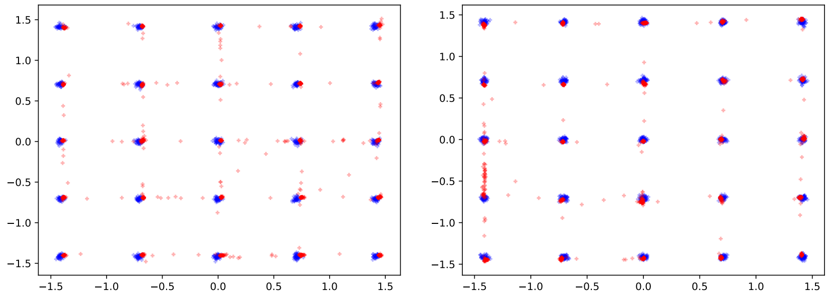
Figure D.2: Fitting the ‘Swiss Roll’ up to 10^5 iterations.



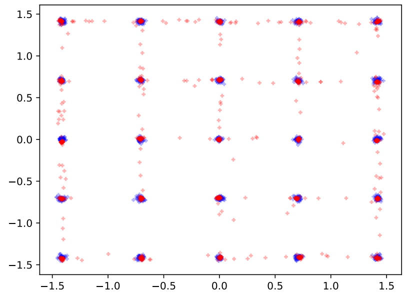
(a) SGD (b) Optimistic Adam



(c) MixedNE-LD

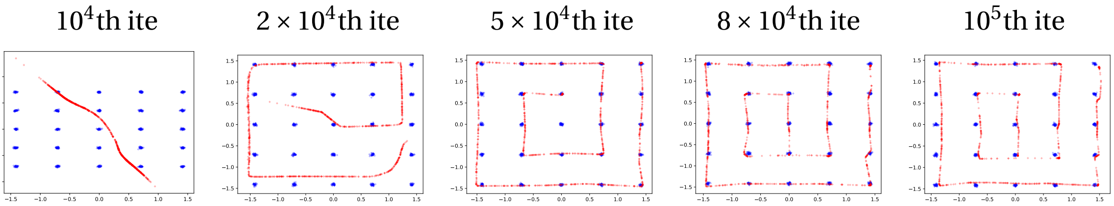


(d) Adam (e) Extra Adam

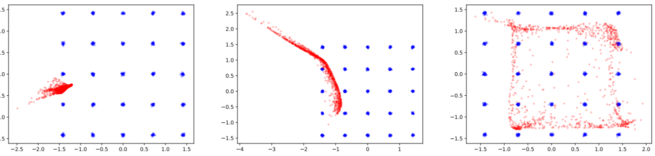


(f) Mirror-Prox-GAN

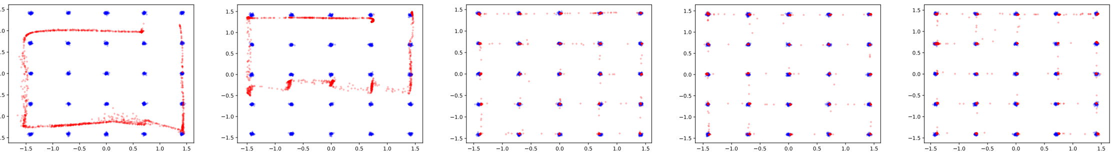
Figure D.3: Fitting 25 Gaussian mixtures up to 10^5 iterations.



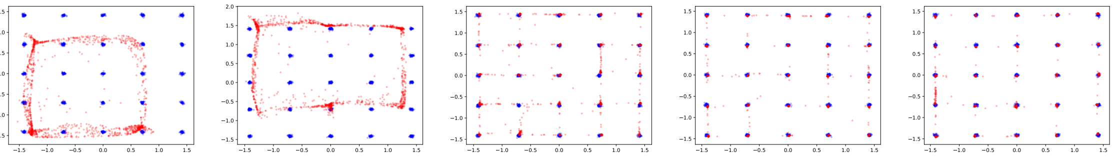
(a) SGD



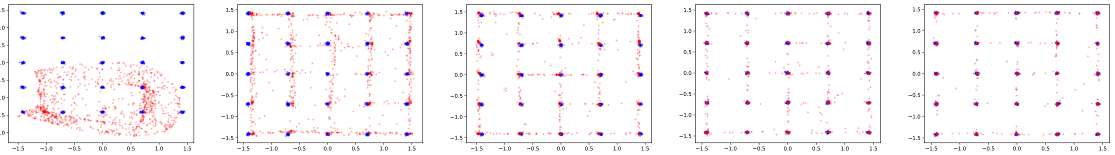
(b) Adam



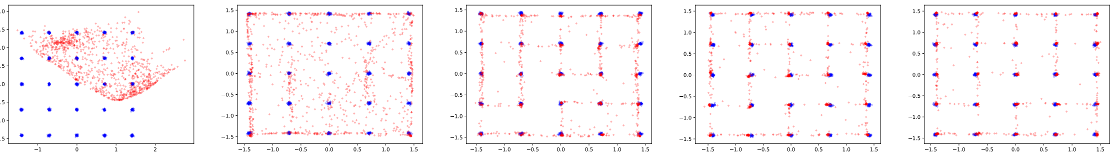
(c) Optimistic Adam



(d) Extra Adam



(e) MixedNE-LD



(f) Mirror-Prox-GAN

Figure D.4: Learning 25 Gaussian mixtures accross different iterations.

D.5.2 Real Data

`MNSIT`

Results on `MNIST` dataset are shown in Fig. D.5. The models are trained by each algorithm for 10^5 iterations. We can see that all algorithms achieve comparable performance. Therefore, the dataset seems too weak to be a discriminator for different algorithms.

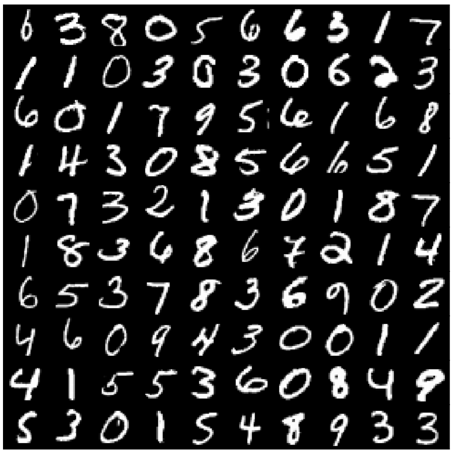
`LSUN Bedroom`

Algorithm	**RMSProp**	**Adam**	**Entropic MD**	**Extra-Adam**
Simultaneous	-	-	3.0955	2.0015
Alternated	3.0555	1.3730	-	3.1620

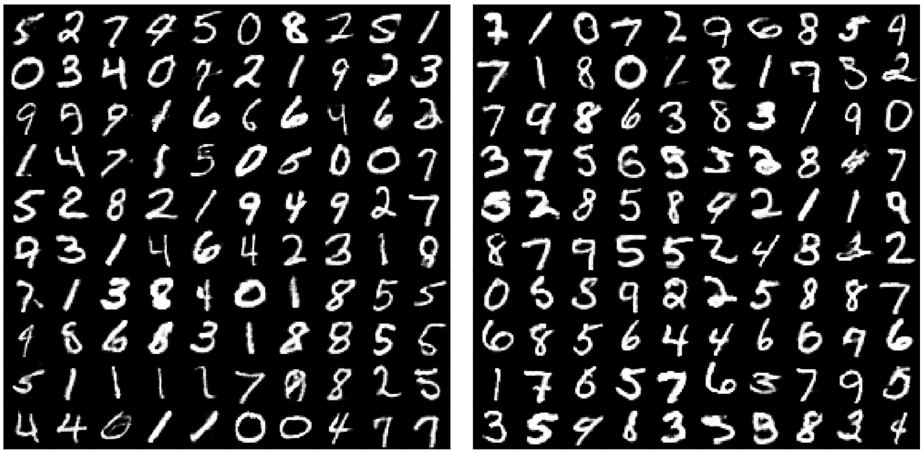
Table D.2: Inception Score of generator trained on LSUN dataset. The reported scores are based on the average of 6400 images from each generator.

More results on the `LSUN bedroom` dataset are shown in Fig. D.6. We show images generated after $4 \times 10^4, 8 \times 10^4$, and 10^5 iterations by each algorithm. We can see that the MixedNE-LD and Alternated Extra-Adam outperform vanilla RMSProp. Adam was able to obtain meaningful images in early stages of training. However, further iterations do not improve the image quality of Adam. In contrast, they lead to severe mode collapse at the 8×10^4th iteration, and converge to noise later on. Simultaneous Extra-Adam completely fails in this task.

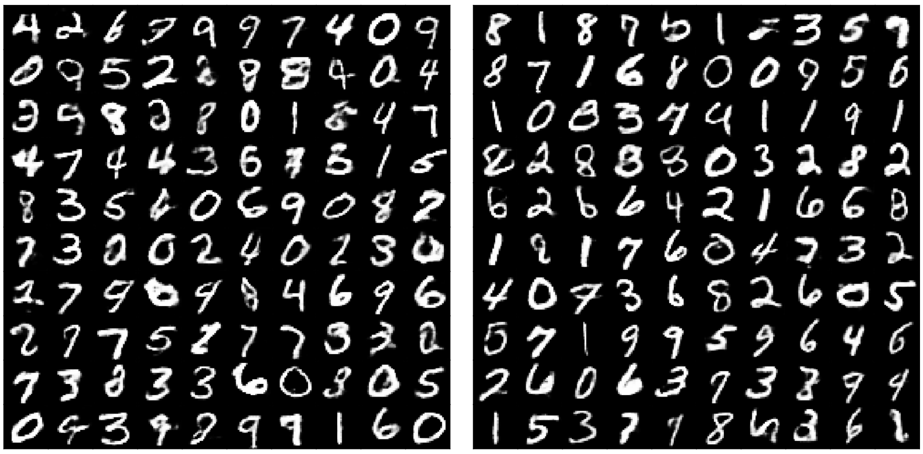
Finally, for reference, we report the Inception Score in Table D.2.



(a) True Data

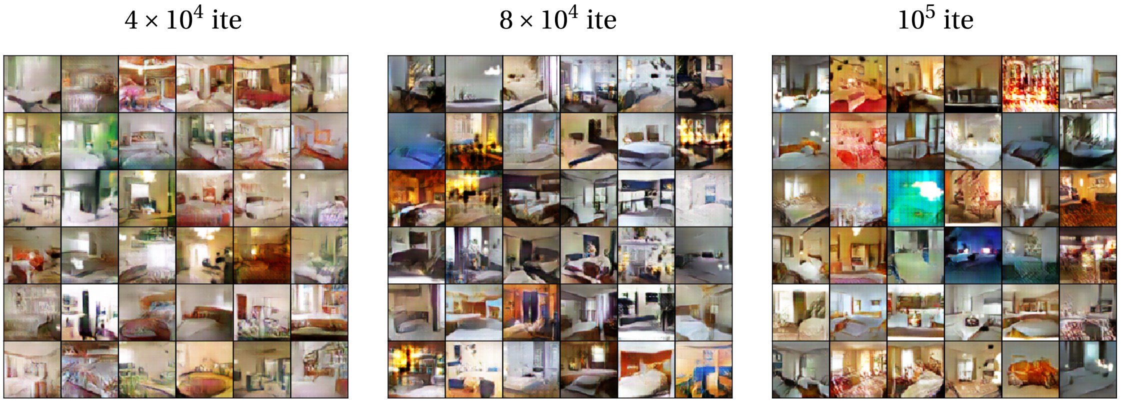


(b) SGD (c) Adam

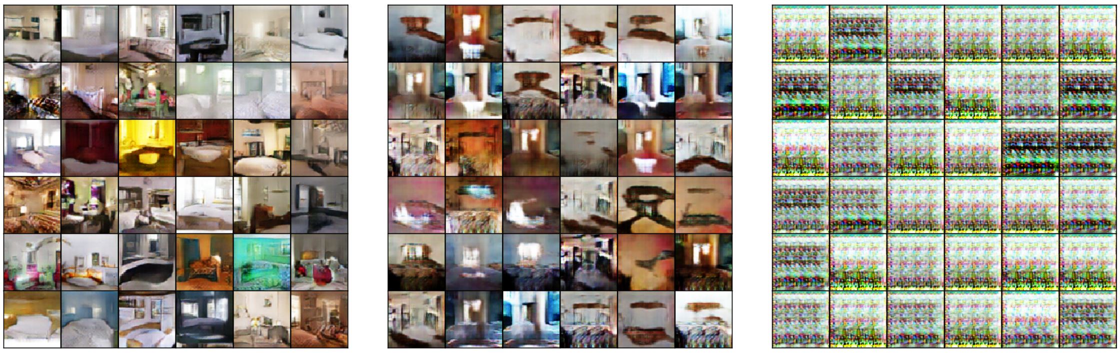


(d) MixedNE-LD (e) Mirror-Prox-GAN

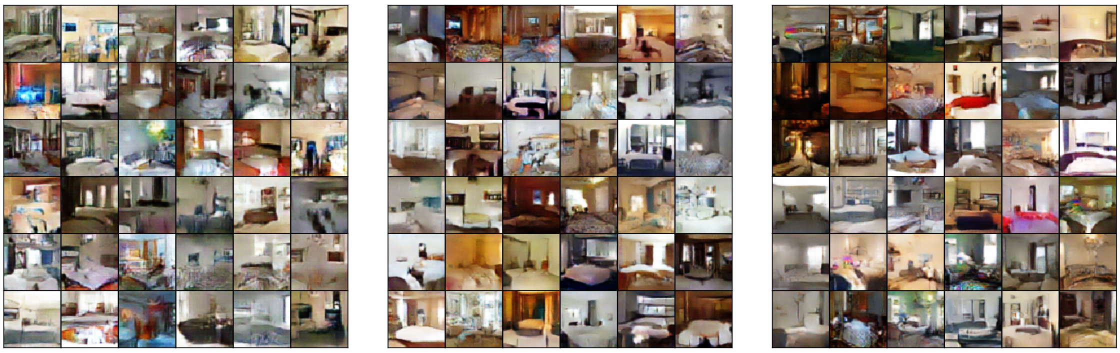
Figure D.5: True `MNIST` images and samples generated by different algorithms.



(a) RMSProp



(b) Adam

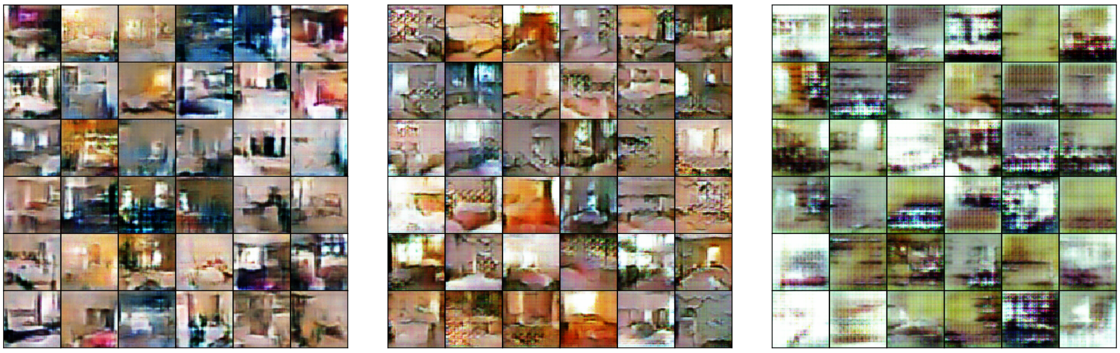


(c) MixedNE-LD, Algorithm 9

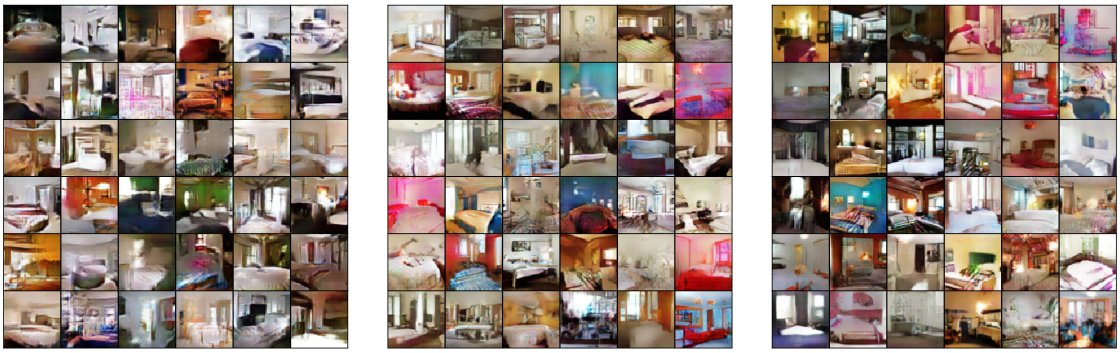
D.6 Details and further results for Section 5.7

D.6.1 Off-policy (DDPG) experiments: algorithms, hyperparameters, and results

- Algorithms:
    1. MixedNE-LD: Algorithm 14
    2. Baselines: Algorithm 15 (with GAD and Extra-Adam)



(d) Simultaneous Extra-Adam



(e) Alternated Extra-Adam

Figure D.6: Image generated by RMSProp, Simultaneous and Alternated Extra-Adam, Adam, and MixedNE-LD on the `LSUN bedroom` dataset.

- Hyperparameters:
    1. Common hyperparameters for Algorithm 14 and Algorithm 15: Table D.3
    2. Exploration-related hyperparameters for Algorithm 14 and Algorithm 15 (the best performing values for every environment are presented): Tables D.4–D.5
- Results:
    1. Heat maps (mass-noise) for NR-MDP setting with $\delta = 0.1$ (Figs. D.7–D.8)
    2. Heat maps (mass-noise) for NR-MDP setting with $\delta = 0$ (Figs. D.9–D.10)
    3. Heat maps (friction-noise) for NR-MDP setting with $\delta = 0.1$ (Figs. D.11–D.12)
    4. Heat maps (friction-noise) for NR-MDP setting with $\delta = 0$ (Figs. D.13–D.14)
    5. Heat maps (mass-friction) for NR-MDP setting with $\delta = 0.1$ (Figs. D.15–D.16)
    6. Heat maps (mass-friction) for NR-MDP setting with $\delta = 0$ (Figs. D.17–D.18)

D.6.2 On-Policy (VPG) experiments: algorithms, and hyperparameters, and results

- Algorithms:
    1. MixedNE-LD: Algorithm 16
    2. Baselines: Algorithm 17 (with GAD and Extra-Adam)
- Hyperparameters:
    1. Common hyperparameters for Algorithm 16 and Algorithm 17: Table D.6
    2. Additional hyperparameters for Algorithm 16 and Algorithm 17 (the best performing values are presented): Tables D.7–D.8
- Results:
    1. NR-MDP setting with $\delta = 0.1$ (Fig. D.19(a))
    2. NR-MDP setting with $\delta = 0$ (Fig. D.19(b))

D.6.3 Ablation study

- Ablation on (β, K_t): see Figs. D.20–D.22.
- Ablation on δ: see Figs. D.23–D.25. If the δ value is way larger (overly conservative) than the requirement (range of environmental changes), it could negatively impact the generalization ability. Choosing the appropriate value of delta is problem dependent.
- HalfCheetah-v2 is trained over 2M steps (cf. Figs. D.28–D.31).

D.6.4 Code

The code repository (for all the experiments): https://github.com/DaDaCheng/LIONS-RL/tree/master/Robust-Reinforcement-Learning-via-Adversarial-training-with-Langevin-Dynamics.

Table D.3: Common hyperparameters for Algorithm 14 and Algorithm 15, where most of the values are chosen from [DHK+17].

Hyperparameter	Value
critic optimizer	Adam
critic learning rate	10^{-3}
target update rate τ	0.999
mini-batch size N	128
discount factor γ	0.99
damping factor β	0.9
replay buffer size	10^6
action noise parameter σ	$\{0, 0.01, 0.1, 0.2, 0.3, 0.4\}$
RMSProp parameter α	0.999
RMSProp parameter ϵ	10^{-8}
RMSProp parameter η	10^{-4}
thermal noise σ_t (Algorithm 14)	$\sigma_0 \times (1 - 5 \times 10^{-5})^t$, where $\sigma_0 \in \left\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\right\}$
warmup steps K_t (Algorithm 14)	$\min\left\{15, \lfloor (1 + 10^{-5})^t \rfloor\right\}$

Table D.4: Exploration-related hyperparameters for Algorithm 14 and Algorithm 15 chosen via grid search (for NR-MDP setting with $\delta = 0.1$).

	Algorithm 14: (σ_0, σ)	Algorithm 15 (with GAD): σ	Algorithm 15 (with Extra-Adam): σ
Walker-v2	$(10^{-2}, 0.01)$	0	0.3
HalfCheetah-v2	$(10^{-2}, 0)$	0.2	0.01
Hopper-v2	$(10^{-3}, 0.2)$	0.2	0.3
Ant-v2	$(10^{-4}, 0.2)$	0.4	0.01
Swimmer-v2	$(10^{-5}, 0.4)$	0.4	0.4
Reacher-v2	$(10^{-3}, 0.2)$	0.4	0.2
Humanoid-v2	$(10^{-4}, 0.01)$	0	0.01
InvertedPendulum-v2	$(10^{-3}, 0.01)$	0.1	0.01

Table D.5: Exploration-related hyperparameters for Algorithm 14 and Algorithm 15 chosen via grid search (for NR-MDP setting with $\delta = 0$).

	Algorithm 14: (σ_0, σ)	Algorithm 15 (with GAD): σ	Algorithm 15 (with Extra-Adam): σ
Walker-v2	$(10^{-2}, 0.1)$	0.01	0.2
HalfCheetah-v2	$(10^{-2}, 0.01)$	0.4	0.01
Hopper-v2	$(10^{-5}, 0.3)$	0.4	0.1
Ant-v2	$(10^{-2}, 0.4)$	0.4	0.01
Swimmer-v2	$(10^{-2}, 0.2)$	0.3	0.3
Reacher-v2	$(10^{-3}, 0.2)$	0.3	0.2
Humanoid-v2	$(10^{-2}, 0.1)$	0	0.01
InvertedPendulum-v2	$(10^{-3}, 0)$	0.01	0.01

Table D.6: Common hyperparameters for Algorithm 16 and Algorithm 17.

Hyperparameter	Value
discount factor γ	0.99
trajectory length H	500
number of trajectories per step $\lvert\mathcal{D}_k\rvert$	1
RMSProp parameter α	0.99
RMSProp parameter ϵ	10^{-8}
learning rate η	$\{10^{-3}, 10^{-4}, 10^{-5}\}$
damping factor β	0.9

Table D.7: Additional hyperparameters for Algorithm 16 and Algorithm 17 chosen via grid search (for NR-MDP setting with $\delta = 0.1$)

	Algorithm 16: (σ_0, η, N_k)	Algorithm 17 (with GAD): η	Algorithm 17 (with Extra-Adam): η
$\rho = 0.2$	$(10^{-5}, 10^{-3}, 1)$	10^{-4}	10^{-4}

Table D.8: Additional hyperparameters for Algorithm 16 and Algorithm 17 chosen via grid search (for NR-MDP setting with $\delta = 0$)

	Algorithm 16: (σ_0, η, N_k)	Algorithm 17 (with GAD): η	Algorithm 17 (with Extra-Adam): η
$\rho = 0.2$	$(10^{-4}, 10^{-4}, 10)$	10^{-4}	10^{-3}

Algorithm 13: Algorithms in Section 5.5 (MixedNE-LD / GAD / EG)

```
Input: step-size $\{\eta_t\}_{t=1}^T$, thermal-noise $\{\epsilon_t\}_{t=1}^T$, warmup steps $\{K_t\}_{t=1}^T$, exponential damping factor $\beta$.
for $t = 1, 2, \ldots, T-1$ do
    MixedNE-LD:
    $\bar{\boldsymbol{y}}_t, \boldsymbol{y}_t^{(1)} \leftarrow \boldsymbol{y}_t$ ; $\bar{\boldsymbol{x}}_t, \boldsymbol{x}_t^{(1)} \leftarrow \boldsymbol{x}_t$
    for $k = 1, 2, \ldots, K_t$ do
        $\xi, \xi' \sim \mathcal{N}(0, I)$
        $\boldsymbol{x}_t^{(k+1)} \leftarrow \Pi_{\boldsymbol{x}}\left(\boldsymbol{x}_t^{(k)} + \eta_t \nabla_{\boldsymbol{x}} F(\boldsymbol{x}_t^{(k)}, \boldsymbol{y}_t) + \epsilon_t \sqrt{2\eta_t}\xi'\right)$
        $\boldsymbol{y}_t^{(k+1)} \leftarrow \Pi_{\boldsymbol{y}}\left(\boldsymbol{y}_t^{(k)} - \eta_t \nabla_{\boldsymbol{y}} F(\boldsymbol{x}_t, \boldsymbol{y}_t^{(k)}) + \epsilon_t \sqrt{2\eta_t}\xi\right)$
        $\bar{\boldsymbol{y}}_t \leftarrow (1-\beta)\bar{\boldsymbol{y}}_t + \beta \boldsymbol{y}_t^{(k+1)}$
        $\bar{\boldsymbol{x}}_t \leftarrow (1-\beta)\bar{\boldsymbol{x}}_t + \beta \boldsymbol{x}_t^{(k+1)}$
    end for
    $\boldsymbol{x}_{t+1} \leftarrow (1-\beta)\boldsymbol{x}_t + \beta \bar{\boldsymbol{x}}_t$
    $\boldsymbol{y}_{t+1} \leftarrow (1-\beta)\boldsymbol{y}_t + \beta \bar{\boldsymbol{y}}_t$

    GAD (Gradient Ascent Descent):
        $\boldsymbol{x}_{t+1} \leftarrow \Pi_{\boldsymbol{x}}\left(\boldsymbol{x}_t + \eta_t \nabla_{\boldsymbol{x}} F(\boldsymbol{x}_t, \boldsymbol{y}_t)\right)$
        $\boldsymbol{y}_{t+1} \leftarrow \Pi_{\boldsymbol{y}}\left(\boldsymbol{y}_t - \eta_t \nabla_{\boldsymbol{y}} F(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t)\right)$

    EG (Extra-Gradient):
        $\boldsymbol{x}_{t+\frac{1}{2}} \leftarrow \Pi_{\boldsymbol{x}}\left(\boldsymbol{x}_t + \eta_t \nabla_{\boldsymbol{x}} F(\boldsymbol{x}_t, \boldsymbol{y}_t))\right)$
        $\boldsymbol{y}_{t+\frac{1}{2}} \leftarrow \Pi_{\boldsymbol{y}}\left(\boldsymbol{y}_t - \eta_t \nabla_{\boldsymbol{y}} F(\boldsymbol{x}_t, \boldsymbol{y}_t)\right)$
        $\boldsymbol{x}_{t+1} \leftarrow \Pi_{\boldsymbol{x}}\left(\boldsymbol{x}_t + \eta_t \nabla_{\boldsymbol{x}} F(\boldsymbol{x}_{t+\frac{1}{2}}, \boldsymbol{y}_{t+\frac{1}{2}})\right)$
        $\boldsymbol{y}_{t+1} \leftarrow \Pi_{\boldsymbol{y}}\left(\boldsymbol{y}_t - \eta_t \nabla_{\boldsymbol{x}} F(\boldsymbol{x}_{t+\frac{1}{2}}, \boldsymbol{y}_{t+\frac{1}{2}})\right)$
end for
Output: $\boldsymbol{y}_T, \boldsymbol{x}_T$.
```

Algorithm 14: DDPG with MixedNE-LD (pre-conditioner = RMSProp)

```
Hyperparameters: see Table D.3
Initialize (randomly) policy parameters $\boldsymbol{x}_1, \boldsymbol{y}_1$, and Q-function parameter $\phi$.
Initialize the target network parameters $\boldsymbol{x}_{\text{targ}} \leftarrow \boldsymbol{x}_1$, $\boldsymbol{y}_{\text{targ}} \leftarrow \boldsymbol{y}_1$, and $\phi_{\text{targ}} \leftarrow \phi$.
Initialize replay buffer $\mathcal{D}$.
Initialize $m \leftarrow \mathbf{0}$ ; $m' \leftarrow \mathbf{0}$.
$t \leftarrow 1$.
repeat
    Observe state $s$, and select actions $a = \mu_{\boldsymbol{y}_t}(s) + \xi$ ; $a' = \nu_{\boldsymbol{x}_t}(s) + \xi'$, where $\xi, \xi' \sim \mathcal{N}(0, \sigma I)$
    Execute the action $\bar{a} = (1-\delta) a + \delta a'$ in the environment.
    Observe reward $r$, next state $s'$, and done signal $d$ to indicate whether $s'$ is terminal.
    Store $(s, \bar{a}, r, s', d)$ in replay buffer $\mathcal{D}$.
    If $s'$ is terminal, reset the environment state.
    if it's time to update then
        for however many updates do
            $\bar{\boldsymbol{x}}_t, \boldsymbol{x}_t^{(1)} \leftarrow \boldsymbol{x}_t$ ; $\bar{\boldsymbol{y}}_t, \boldsymbol{y}_t^{(1)} \leftarrow \boldsymbol{y}_t$
            for $k = 1, 2, \ldots, K_t$ do
                Sample a random minibatch of $N$ transitions $B = \{(s, \bar{a}, r, s', d)\}$ from $\mathcal{D}$.
                Compute targets $y(r, s', d) = r + \gamma (1-d) Q_{\phi_{\text{targ}}}\left(s', (1-\delta)\mu_{\boldsymbol{y}_{\text{targ}}}(s') + \delta \nu_{\boldsymbol{x}_{\text{targ}}}(s')\right)$.
                Update critic by one step of (preconditioned) gradient descent using $\nabla_\phi L(\phi)$, where
                    $L(\phi) = \frac{1}{N} \sum_{(s, \bar{a}, r, s', d) \in B} \left(y(r, s', d) - Q_\phi(s, \bar{a})\right)^2.$
                Compute the (agent and adversary) policy gradient estimates:
                    $\widehat{\nabla_{\boldsymbol{y}} F(\boldsymbol{y}, \boldsymbol{x}_t)} = \frac{1-\delta}{N} \sum_{s \in \mathcal{D}} \nabla_{\boldsymbol{y}} \mu_{\boldsymbol{y}}(s) \nabla_{\bar{a}} Q_\phi(s, \bar{a}) |_{\bar{a} = (1-\delta)\mu_{\boldsymbol{y}}(s) + \delta \nu_{\boldsymbol{x}_t}(s)}$
                    $\widehat{\nabla_{\boldsymbol{x}} F(\boldsymbol{y}_t, \boldsymbol{x})} = \frac{\delta}{N} \sum_{s \in \mathcal{D}} \nabla_{\boldsymbol{x}} \nu_{\boldsymbol{x}}(s) \nabla_{\bar{a}} Q_\phi(s, \bar{a}) |_{\bar{a} = (1-\delta)\mu_{\boldsymbol{y}_t}(s) + \delta \nu_{\boldsymbol{x}}(s)}.$
                $g \leftarrow \left[\widehat{\nabla_{\boldsymbol{y}} F(\boldsymbol{y}, \boldsymbol{x}_t)}\right]_{\boldsymbol{y} = \boldsymbol{y}_t^{(k)}}$ ; $m \leftarrow \alpha m + (1-\alpha) g \odot g$ ; $C \leftarrow \operatorname{diag}\left(\sqrt{m + \epsilon}\right)$
                $\boldsymbol{y}_t^{(k+1)} \leftarrow \boldsymbol{y}_t^{(k)} + \eta C^{-1} g + \sqrt{2\eta}\sigma_t C^{-\frac{1}{2}} \xi$, where $\xi \sim \mathcal{N}(0, I)$
                $g' \leftarrow \left[\widehat{\nabla_{\boldsymbol{x}} F(\boldsymbol{y}_t, \boldsymbol{x})}\right]_{\boldsymbol{x} = \boldsymbol{x}_t^{(k)}}$ ; $m' \leftarrow \alpha m' + (1-\alpha) g' \odot g'$ ; $D \leftarrow \operatorname{diag}\left(\sqrt{m' + \epsilon}\right)$
                $\boldsymbol{x}_t^{(k+1)} \leftarrow \boldsymbol{x}_t^{(k)} - \eta D^{-1} g' + \sqrt{2\eta}\sigma_t D^{-\frac{1}{2}} \xi'$, where $\xi' \sim \mathcal{N}(0, I)$
                $\bar{\boldsymbol{x}}_t \leftarrow (1-\beta)\bar{\boldsymbol{x}}_t + \beta \boldsymbol{x}_t^{(k+1)}$ ; $\bar{\boldsymbol{y}}_t \leftarrow (1-\beta)\bar{\boldsymbol{y}}_t + \beta \boldsymbol{y}_t^{(k+1)}$
                Update the target networks:
                    $\phi_{\text{targ}} \leftarrow \tau \phi_{\text{targ}} + (1-\tau)\phi$
                    $\boldsymbol{y}_{\text{targ}} \leftarrow \tau \boldsymbol{y}_{\text{targ}} + (1-\tau)\boldsymbol{y}_t^{(k+1)}$
                    $\boldsymbol{x}_{\text{targ}} \leftarrow \tau \boldsymbol{x}_{\text{targ}} + (1-\tau)\boldsymbol{x}_t^{(k+1)}$
            end for
            $\boldsymbol{x}_{t+1} \leftarrow (1-\beta)\boldsymbol{x}_t + \beta \bar{\boldsymbol{x}}_t$ ; $\boldsymbol{y}_{t+1} \leftarrow (1-\beta)\boldsymbol{y}_t + \beta \bar{\boldsymbol{y}}_t$
            $t \leftarrow t + 1$.
        end for
    end if
until convergence
Output: $\boldsymbol{x}_T, \boldsymbol{y}_T$.
```

Algorithm 15: DDPG with GAD (pre-conditioner = RMSProp) / Extra-Adam

```
    Hyperparameters: see Table D.3
    Initialize (randomly) policy parameters $\boldsymbol{x}_1, \boldsymbol{y}_1$, and Q-function parameter $\phi$.
    Initialize the target network parameters $\boldsymbol{x}_{\text{targ}} \leftarrow \boldsymbol{x}_1$, $\boldsymbol{y}_{\text{targ}} \leftarrow \boldsymbol{y}_1$, and $\phi_{\text{targ}} \leftarrow \phi$.
    Initialize replay buffer $\mathcal{D}$.
    Initialize $m \leftarrow \mathbf{0}$ ; $m' \leftarrow \mathbf{0}$.
    $t \leftarrow 1$.
    repeat
        Observe state $s$, and select actions $a = \mu_{\boldsymbol{y}_t}(s) + \xi$ ; $a' = \nu_{\boldsymbol{x}_t}(s) + \xi'$, where $\xi, \xi' \sim \mathcal{N}(0, \sigma I)$
        Execute the action $\bar{a} = (1-\delta)a + \delta a'$ in the environment.
        Observe reward $r$, next state $s'$, and done signal $d$ to indicate whether $s'$ is terminal.
        Store $\left(s, \bar{a}, r, s', d\right)$ in replay buffer $\mathcal{D}$.
        If $s'$ is terminal, reset the environment state.
        if it's time to update then
            for however many updates do
                Sample a random minibatch of $N$ transitions $B = \left\{\left(s, \bar{a}, r, s', d\right)\right\}$ from $\mathcal{D}$.
                Compute targets $y\left(r, s', d\right) = r + \gamma(1-d)\, Q_{\phi_{\text{targ}}}\left(s', (1-\delta)\mu_{\boldsymbol{y}_{\text{targ}}}\left(s'\right) + \delta \nu_{\boldsymbol{x}_{\text{targ}}}\left(s'\right)\right)$.
                Update critic by one step of (preconditioned) gradient descent using $\nabla_\phi L\left(\phi\right)$, where
                    $$L\left(\phi\right) = \frac{1}{N} \sum_{(s,\bar{a},r,s',d) \in B} \left(y\left(r, s', d\right) - Q_\phi(s, \bar{a})\right)^2.$$
                Compute the (agent and adversary) policy gradient estimates:
                    $$\widehat{\nabla_{\boldsymbol{y}} F\left(\boldsymbol{y}, \boldsymbol{x}_t\right)} = \frac{1-\delta}{N} \sum_{s \in \mathcal{D}} \nabla_{\boldsymbol{y}} \mu_{\boldsymbol{y}}(s) \nabla_{\bar{a}} Q_\phi(s, \bar{a})\,|_{\bar{a}=(1-\delta)\mu_{\boldsymbol{y}}(s)+\delta\nu_{\boldsymbol{x}_t}(s)}$$
                    $$\widehat{\nabla_{\boldsymbol{x}} F\left(\boldsymbol{y}_t, \boldsymbol{x}\right)} = \frac{\delta}{N} \sum_{s \in \mathcal{D}} \nabla_{\boldsymbol{x}} \nu_{\boldsymbol{x}}(s) \nabla_{\bar{a}} Q_\phi(s, \bar{a})\,|_{\bar{a}=(1-\delta)\mu_{\boldsymbol{y}_t}(s)+\delta\nu_{\boldsymbol{x}}(s)}.$$
                GAD (pre-conditioner = RMSProp):
                $g \leftarrow \left[\widehat{\nabla_{\boldsymbol{y}} F\left(\boldsymbol{y}, \boldsymbol{x}_t\right)}\right]_{\boldsymbol{y}=\boldsymbol{y}_t}$ ; $m \leftarrow \alpha m + (1-\alpha)\, g \odot g$ ; $C \leftarrow \operatorname{diag}\left(\sqrt{m+\epsilon}\right)$
                $\boldsymbol{y}_{t+1} \leftarrow \boldsymbol{y}_t + \eta C^{-1} g$
                $g' \leftarrow \left[\widehat{\nabla_{\boldsymbol{x}} F\left(\boldsymbol{y}_t, \boldsymbol{x}\right)}\right]_{\boldsymbol{x}=\boldsymbol{x}_t}$ ; $m' \leftarrow \alpha m' + (1-\alpha)\, g' \odot g'$ ; $D \leftarrow \operatorname{diag}\left(\sqrt{m'+\epsilon}\right)$
                $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t - \eta D^{-1} g'$
                Extra-Adam: use Algorithm 4 from [GBV+19].
                Update the target networks:
                    $$\begin{aligned} \phi_{\text{targ}} &\leftarrow \tau \phi_{\text{targ}} + (1-\tau)\phi \\ \boldsymbol{y}_{\text{targ}} &\leftarrow \tau \boldsymbol{y}_{\text{targ}} + (1-\tau)\boldsymbol{y}_{t+1} \\ \boldsymbol{x}_{\text{targ}} &\leftarrow \tau \boldsymbol{x}_{\text{targ}} + (1-\tau)\boldsymbol{x}_{t+1} \end{aligned}$$
                $t \leftarrow t+1$.
            end for
        end if
    until convergence
    Output: $\boldsymbol{x}_T, \boldsymbol{y}_T$.
```

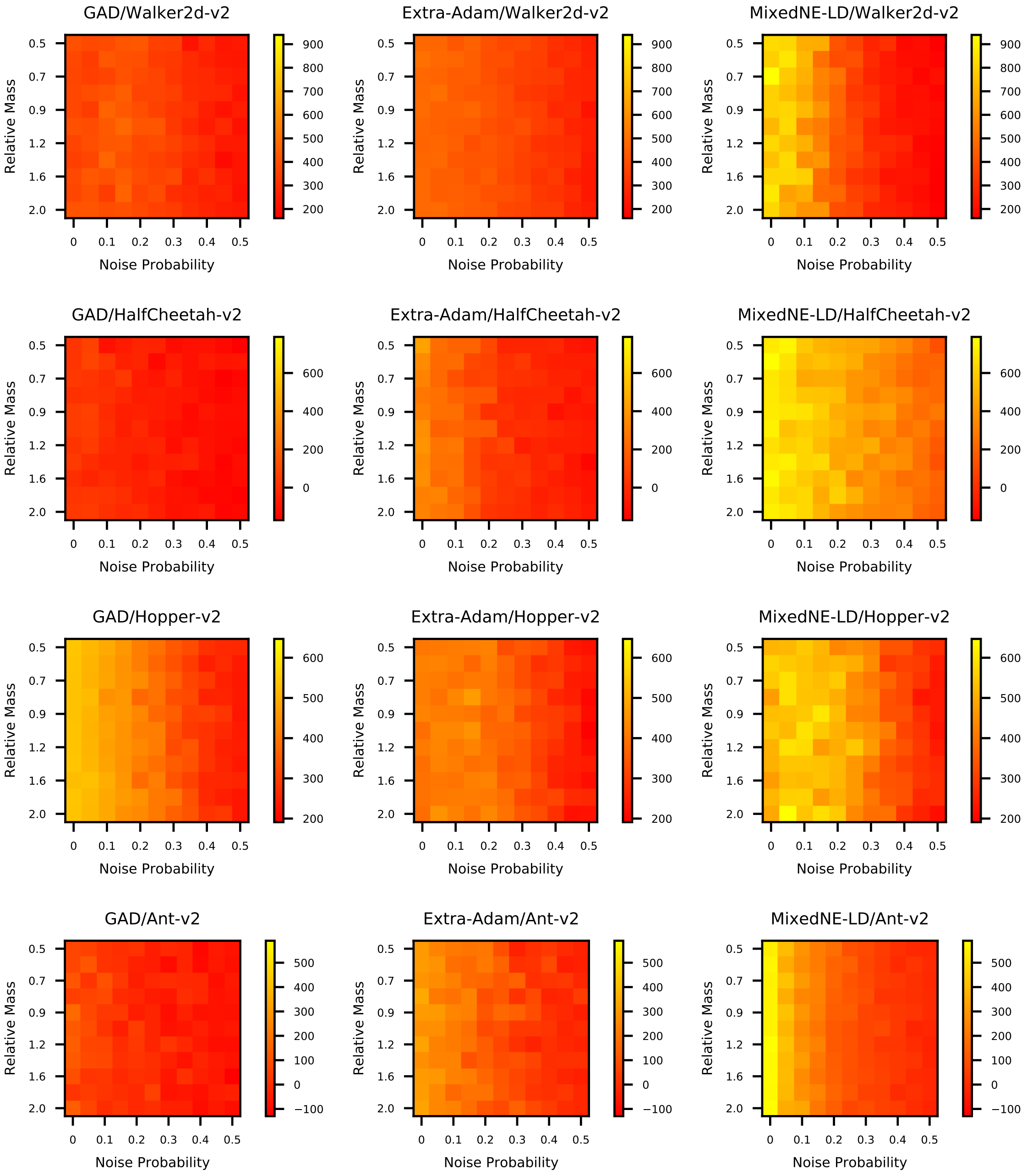


Figure D.7: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of noise probability and mass values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant.

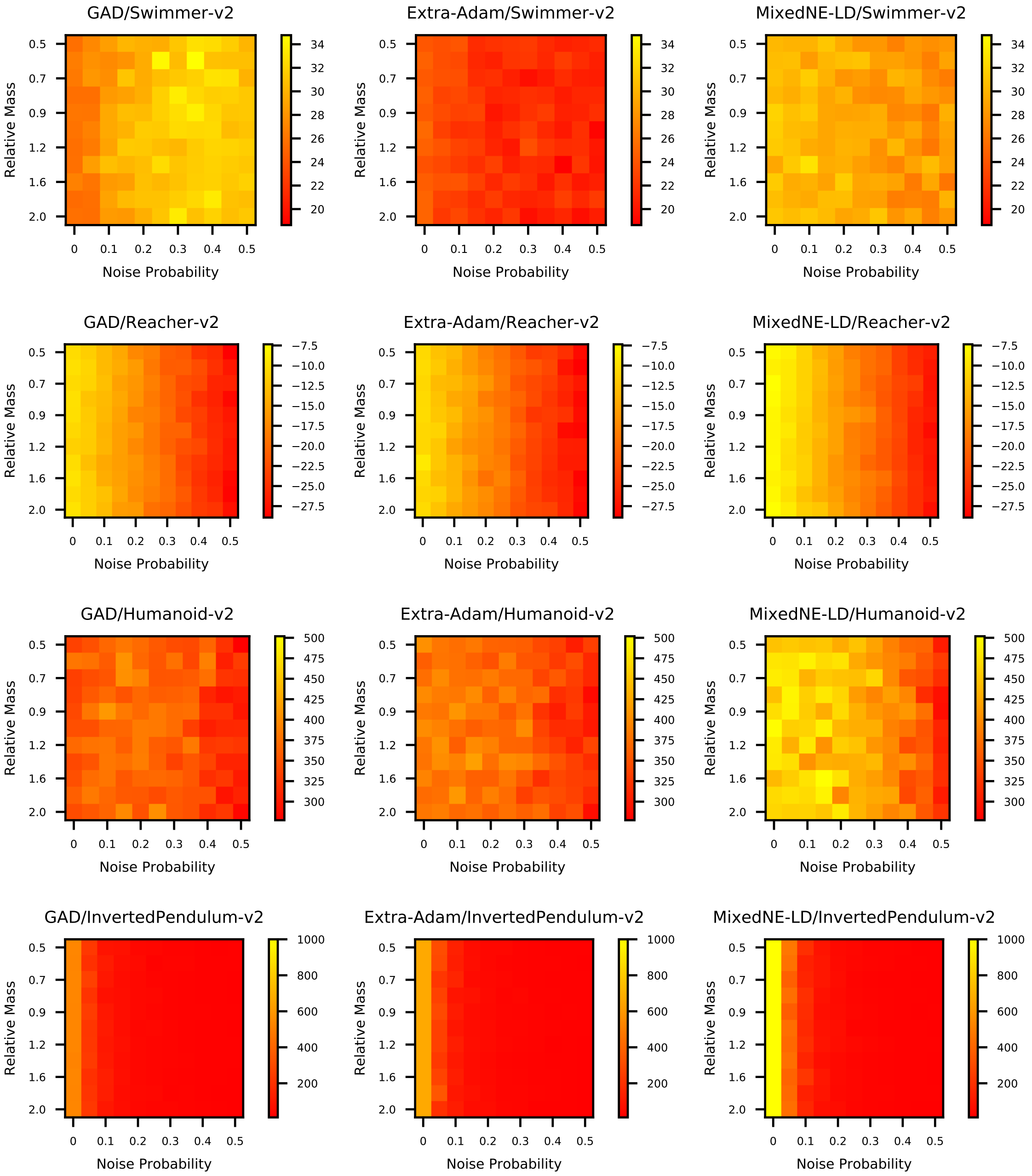


Figure D.8: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of noise probability and mass values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and InvertedPendulum.

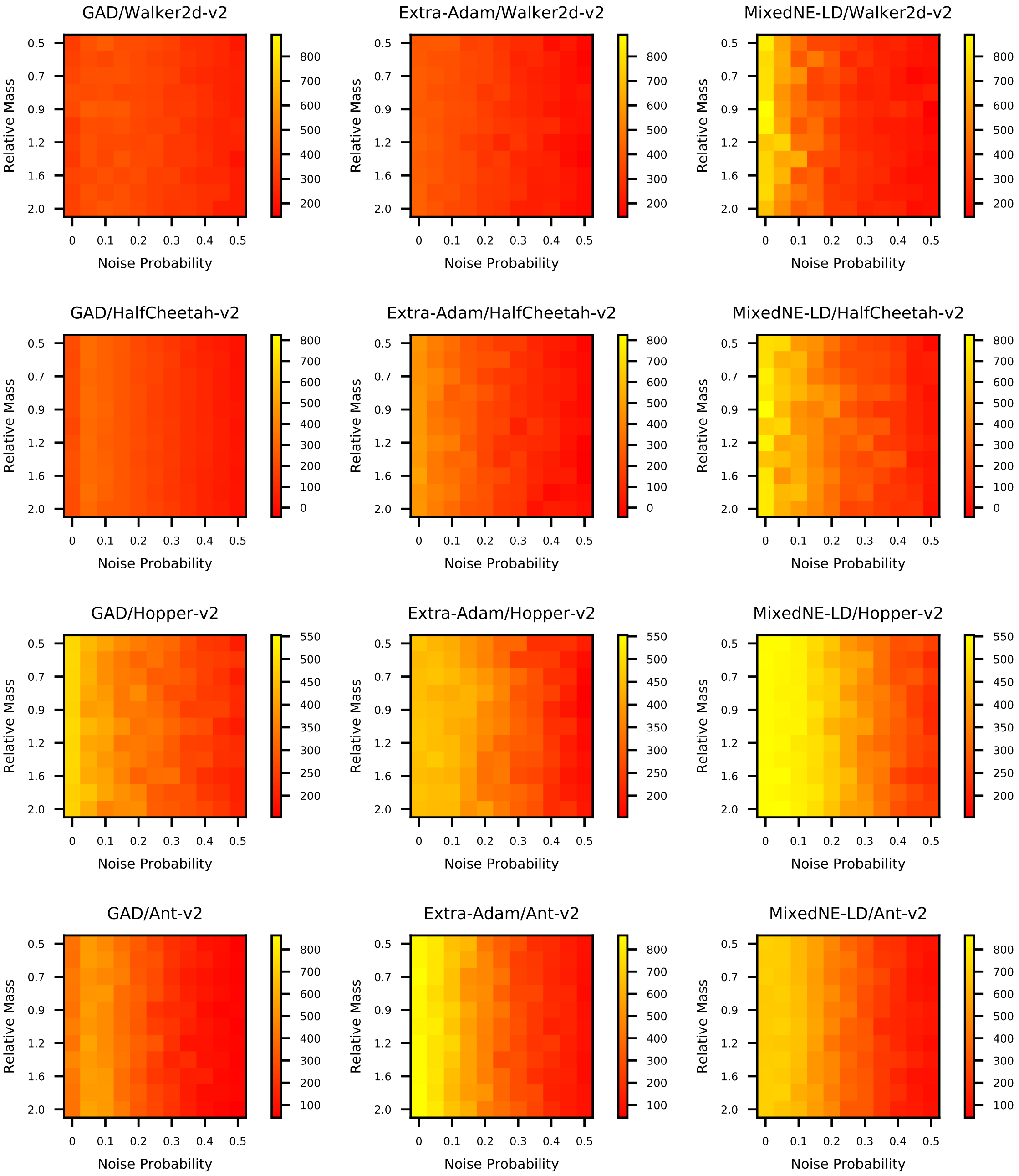


Figure D.9: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of noise probability and mass values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant.

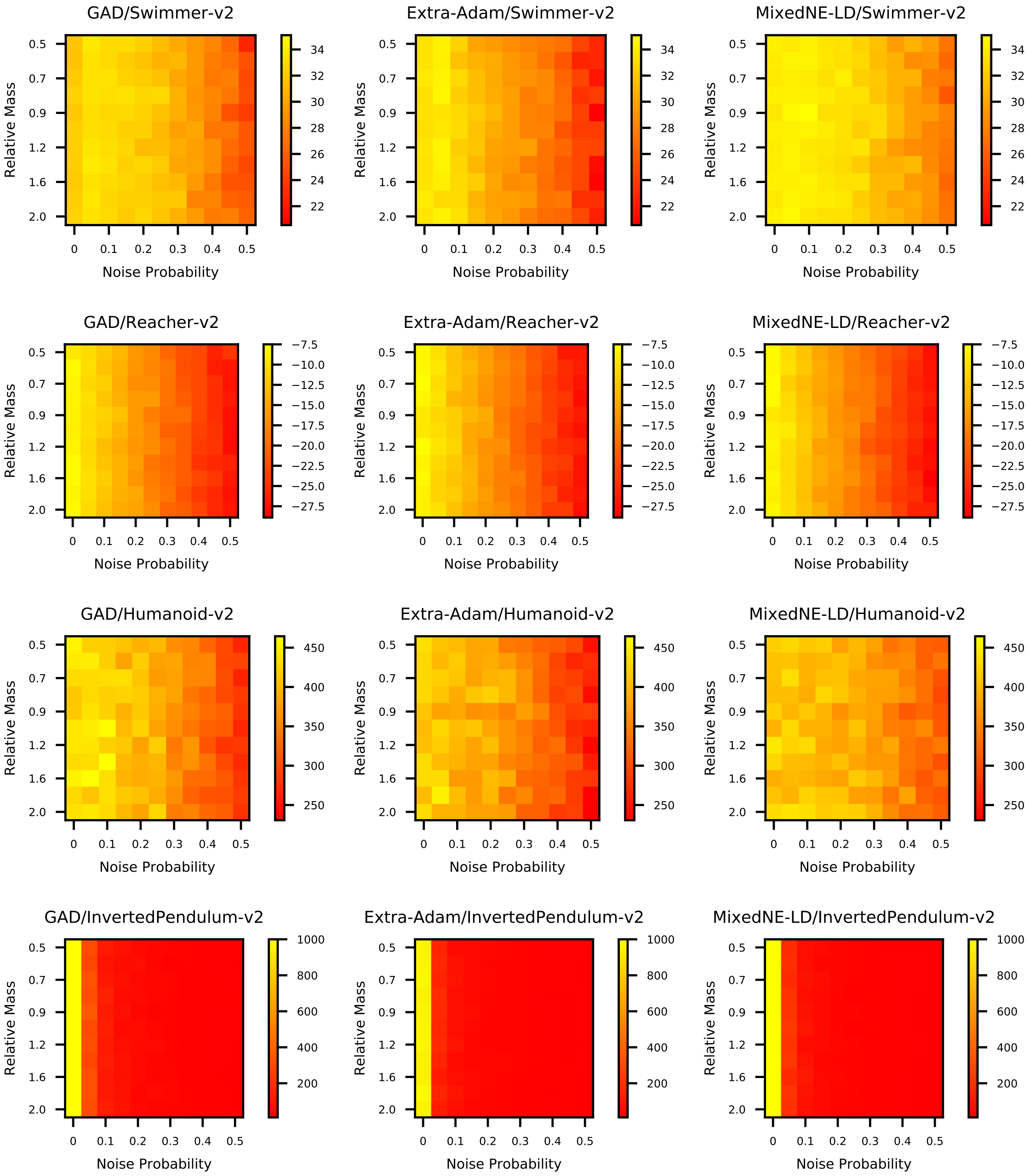
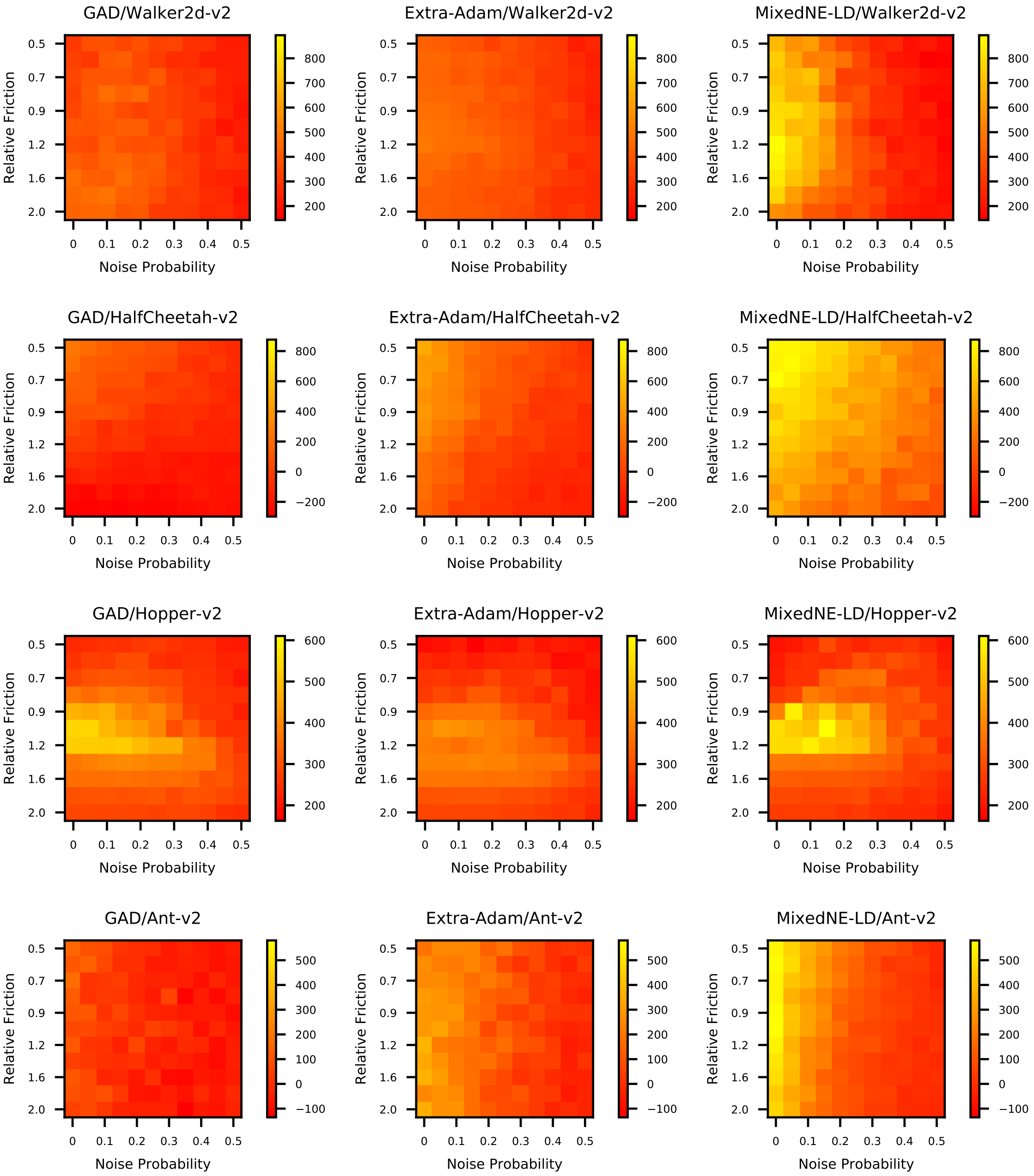


Figure D.10: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of noise probability and mass values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and InvertedPendulum.



GAD/Walker2d-v2

Extra-Adam/Walker2d-v2

MixedNE-LD/Walker2d-v2

GAD/HalfCheetah-v2

Extra-Adam/HalfCheetah-v2

MixedNE-LD/HalfCheetah-v2

GAD/Hopper-v2

Extra-Adam/Hopper-v2

MixedNE-LD/Hopper-v2

GAD/Ant-v2

Extra-Adam/Ant-v2

MixedNE-LD/Ant-v2

Figure D.11: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of noise probability and friction values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant.

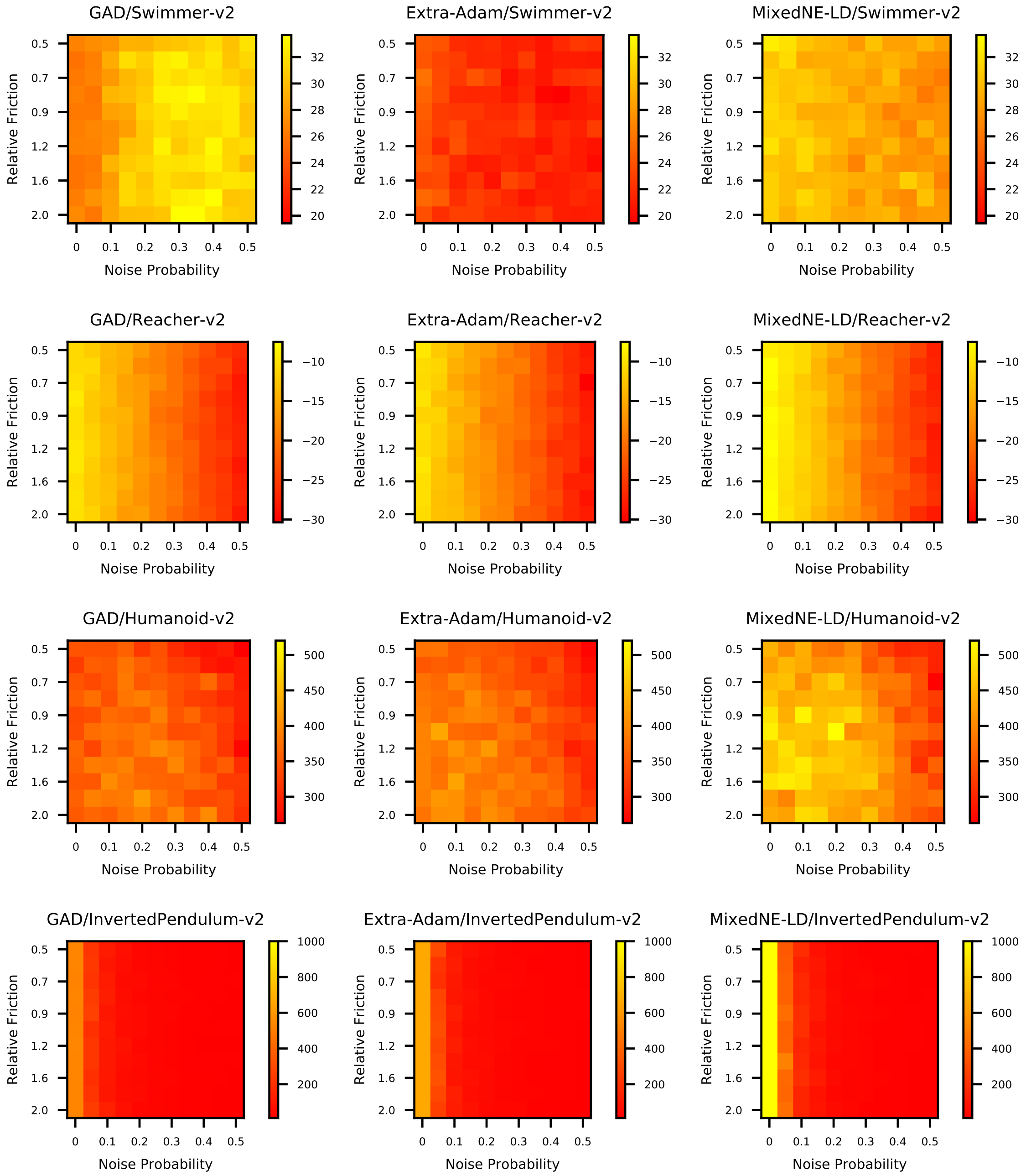


Figure D.12: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of noise probability and friction values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and InvertedPendulum.

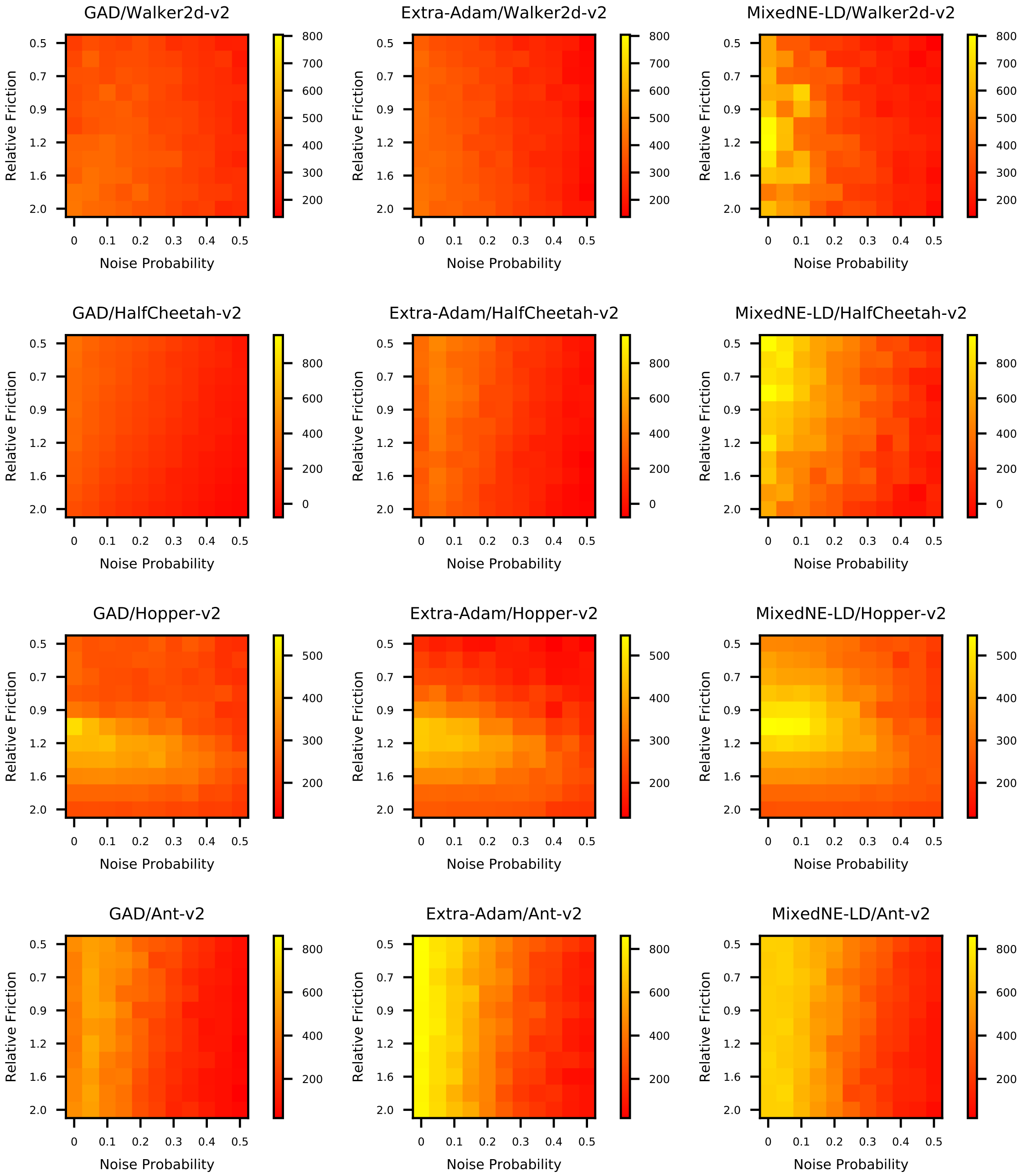


Figure D.13: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of noise probability and friction values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant.

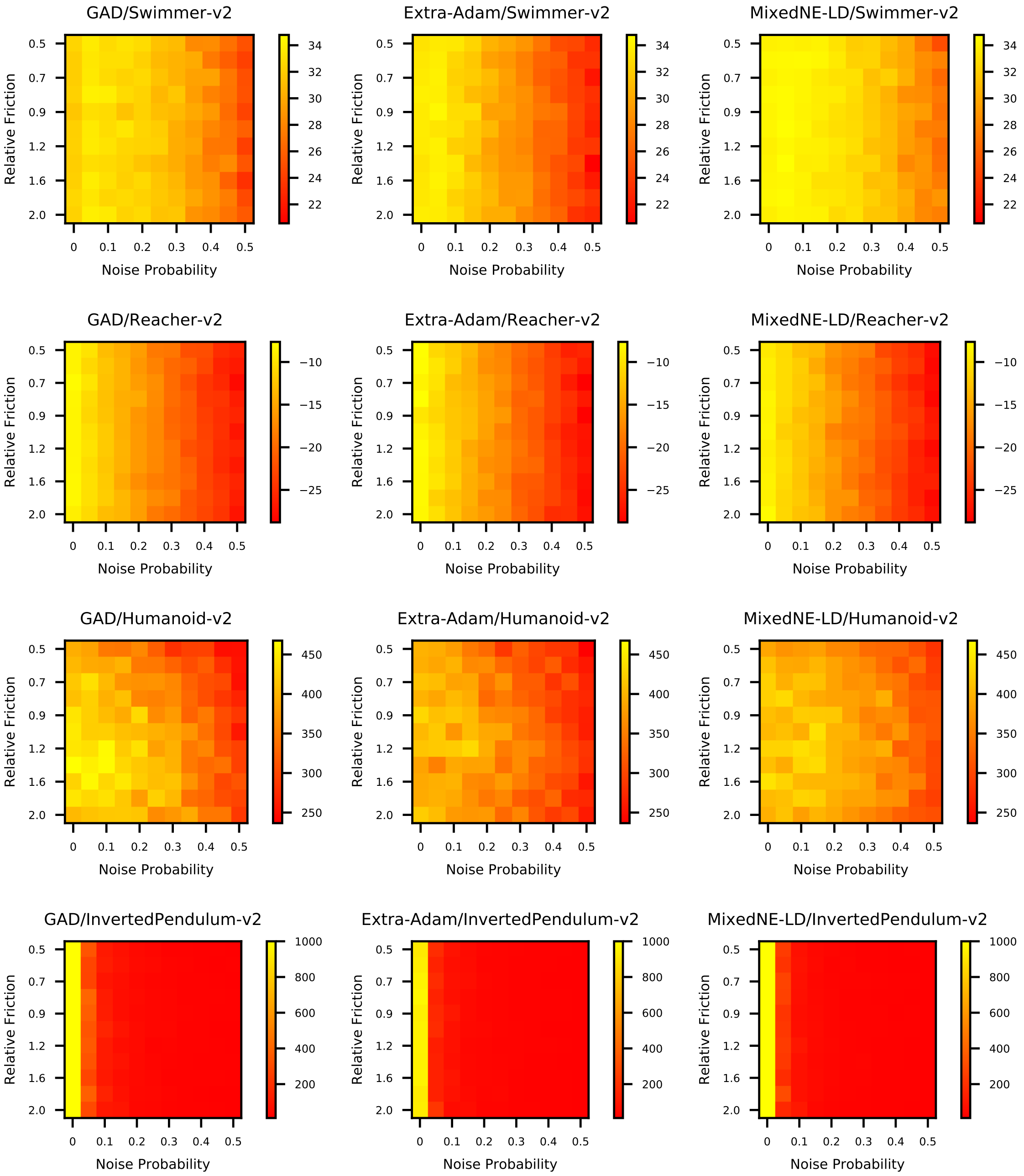


Figure D.14: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of noise probability and friction values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and InvertedPendulum.

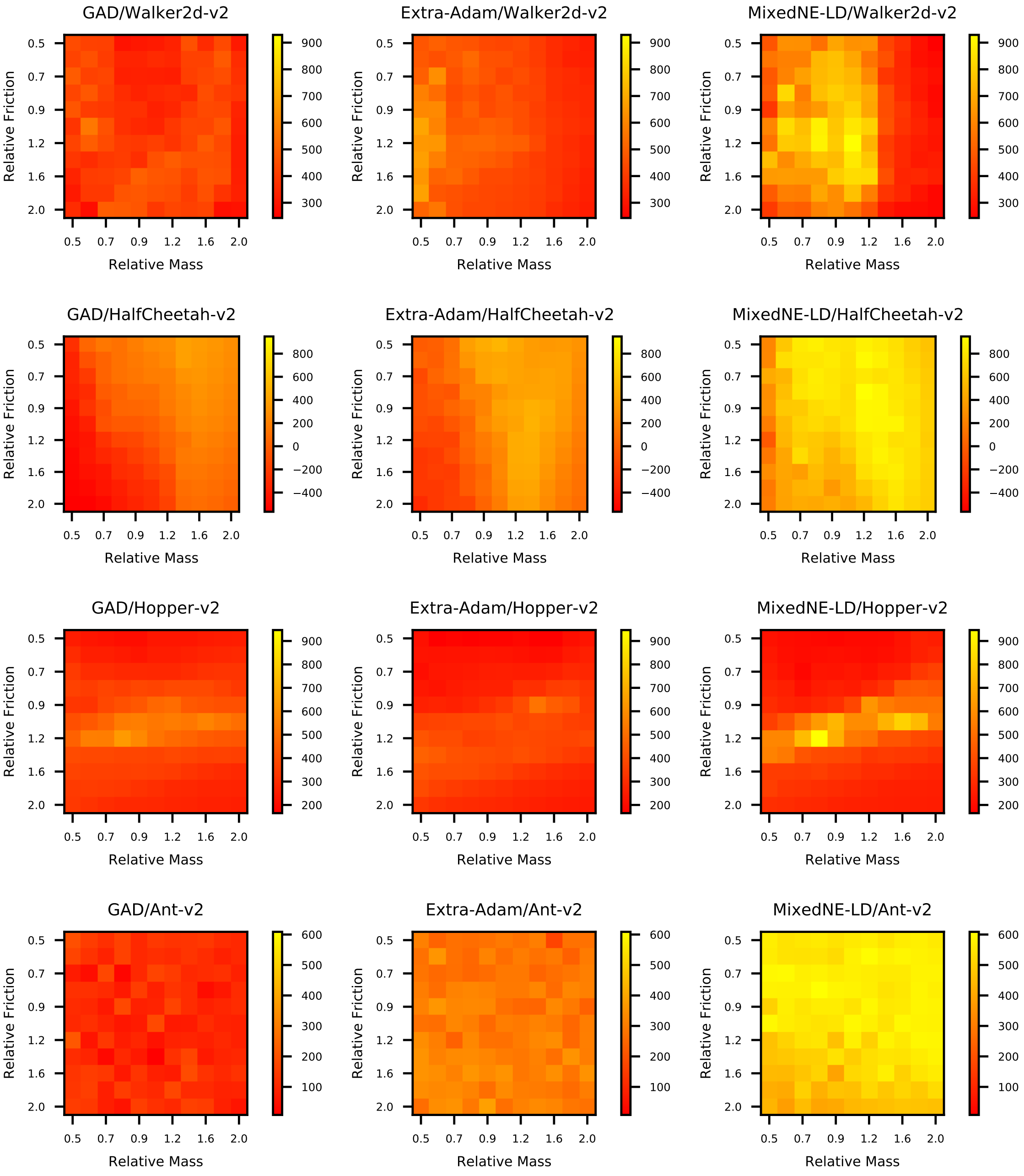


Figure D.15: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of friction and mass values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant.

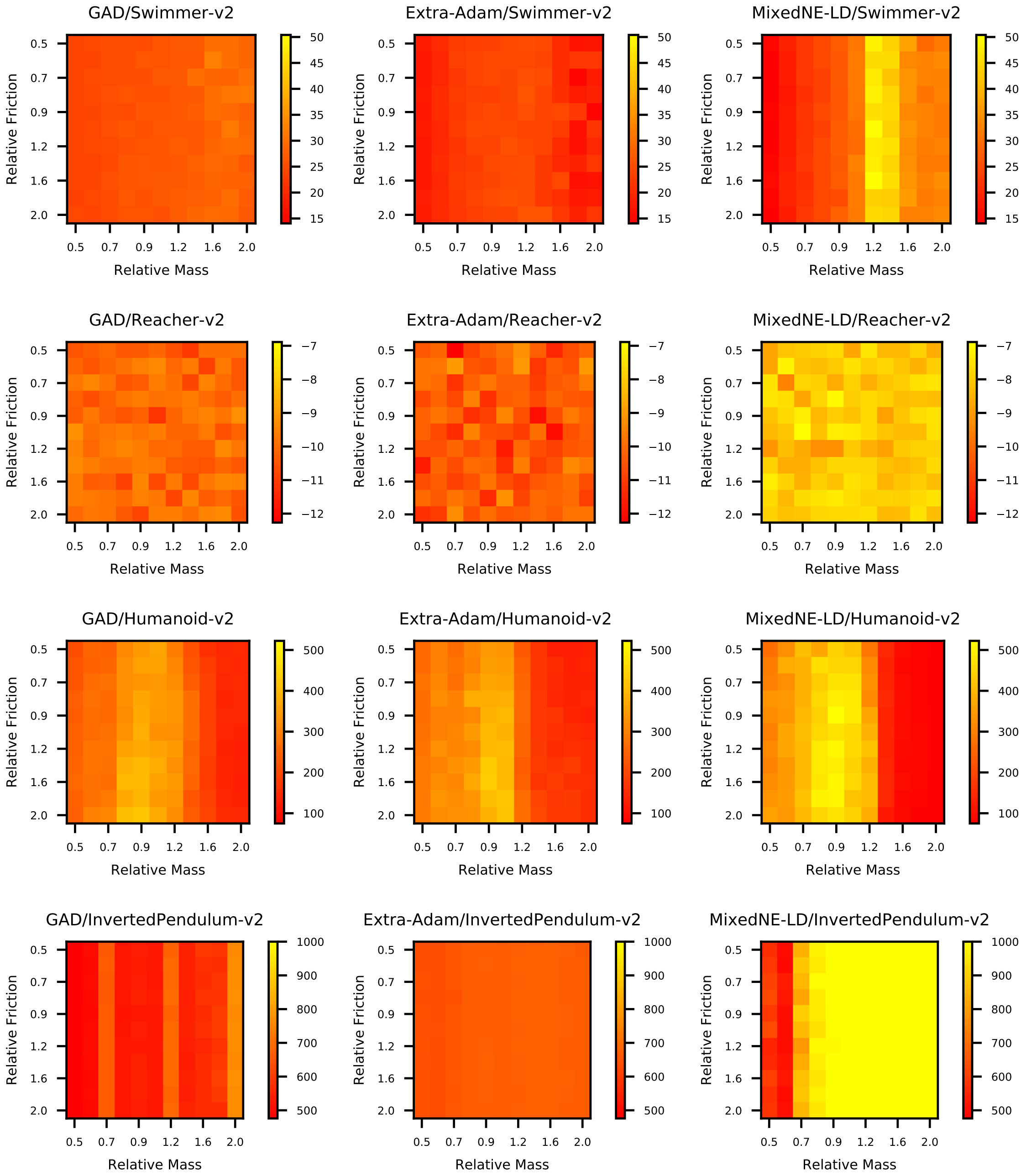


Figure D.16: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of friction and mass values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and InvertedPendulum.

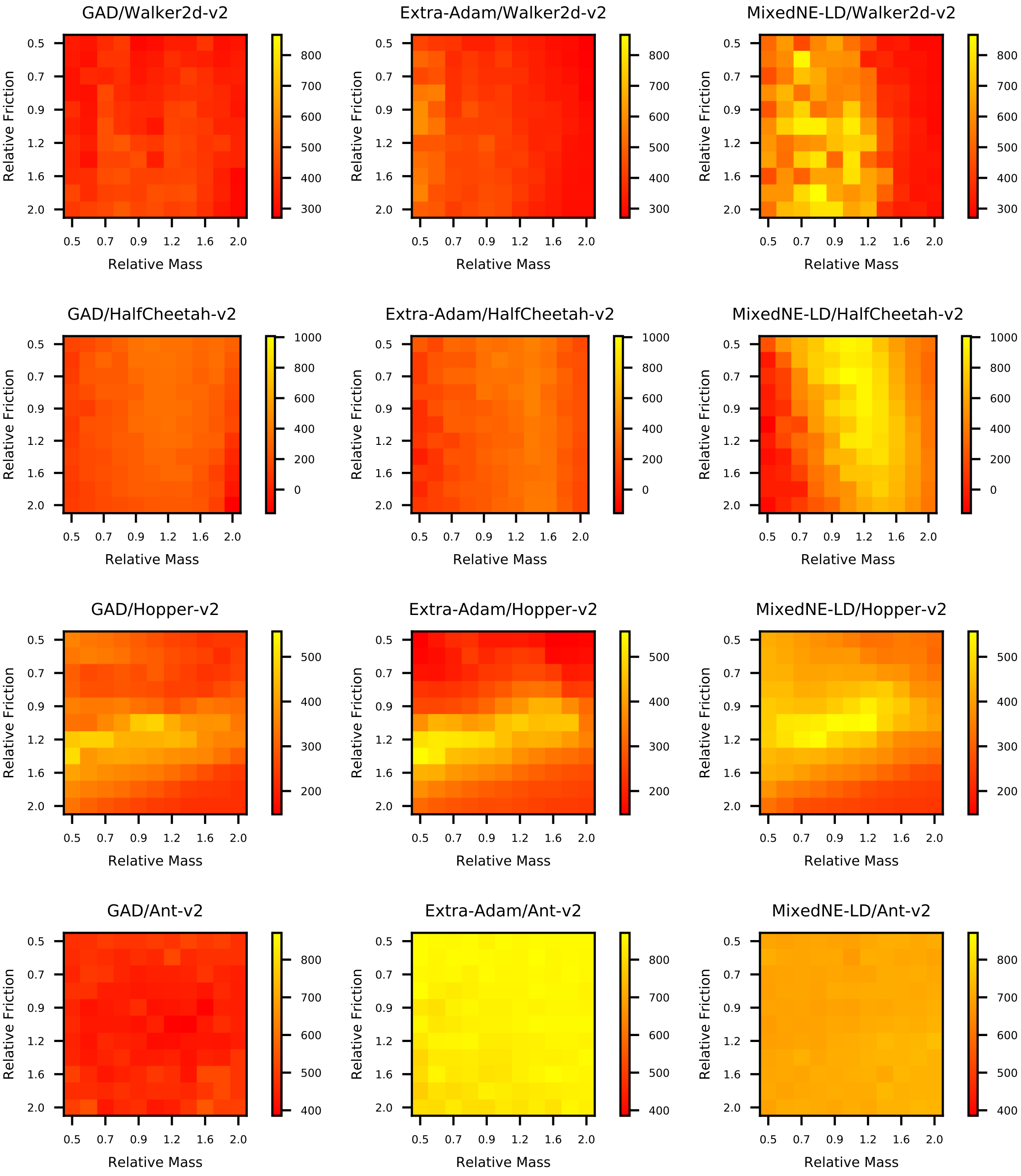


Figure D.17: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of friction and mass values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant.

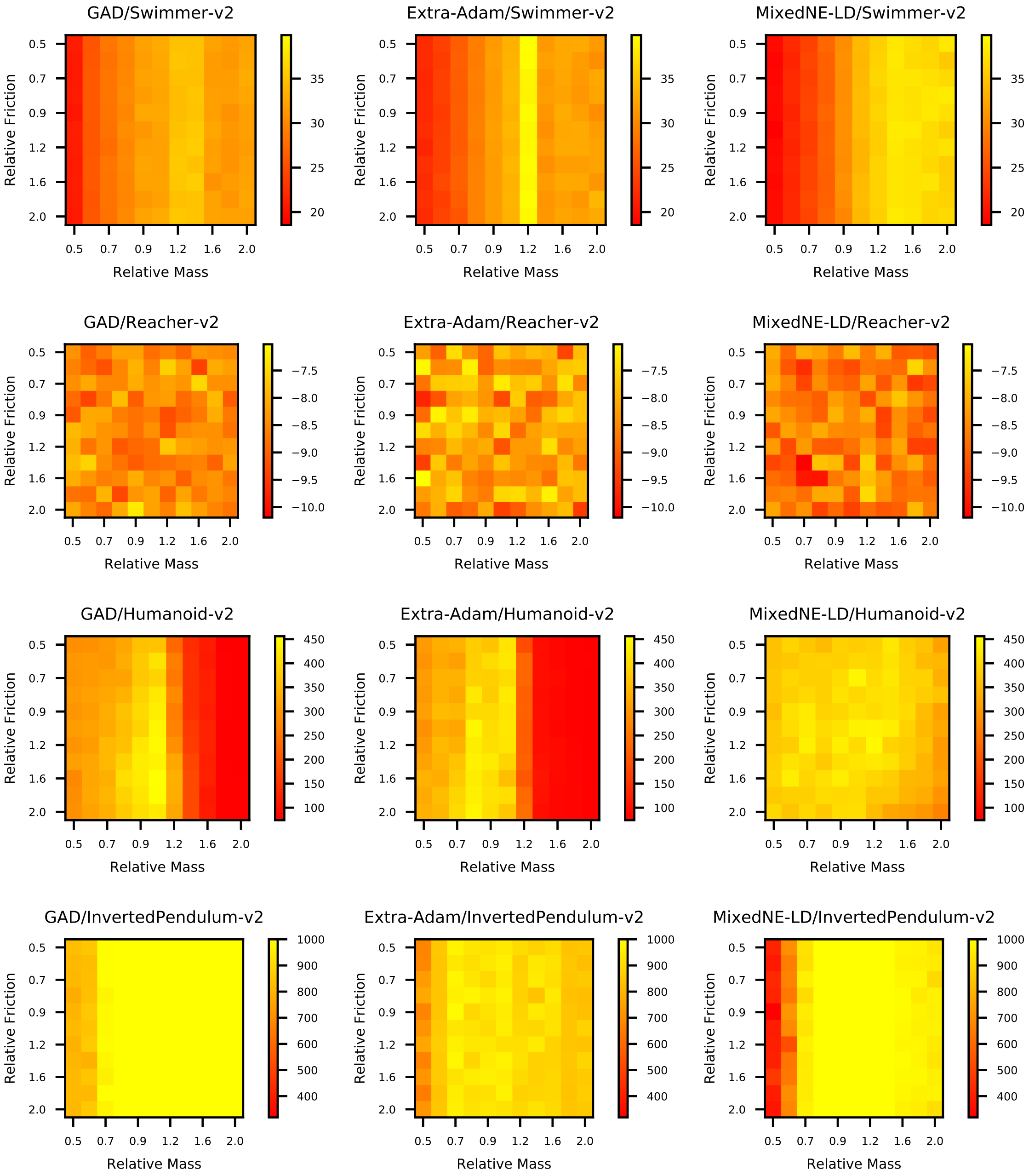


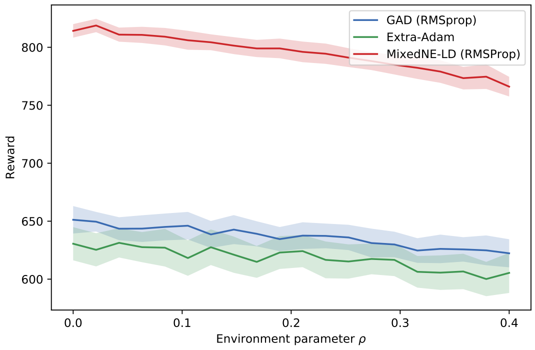
Figure D.18: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of friction and mass values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and InvertedPendulum.

Algorithm 16: VPG with MixedNE-LD (pre-conditioner = RMSProp)

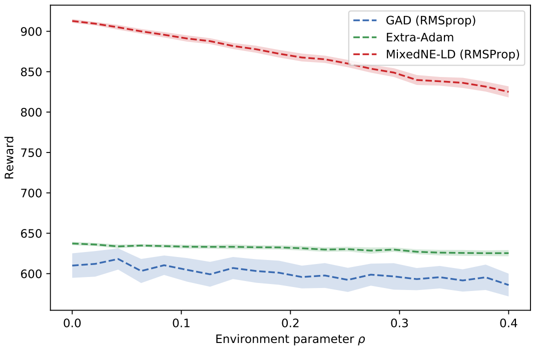
```
Hyperparameters: see Table D.6
Initialize (randomly) policy parameters $\boldsymbol{y}_0$, $w_0$
for $k = 0, 1, 2, \dots$ do
    $\bar{\boldsymbol{y}}_k, \boldsymbol{y}_k^{(0)} \leftarrow \boldsymbol{y}_k$ ; $\bar{w}_k, w_k^{(0)} \leftarrow w_k$
    for $n = 0, 1, \dots, N_k$ do
        Collect set of trajectories $\mathcal{D}_k^{(n)} = \{(\dots, s_t^{(\tau)}, \bar{a}_t^{(\tau)}, r_t^{(\tau)}, \dots)\}_\tau$ by running $\pi_{\boldsymbol{y}_k^{(n)}}$, and $\pi'_{w_k^{(n)}}$ in $\mathcal{M}$, i.e., $a_t \sim \pi_{\boldsymbol{y}_k^{(n)}}(s_t)$, $a'_t \sim \pi'_{w_k^{(n)}}(s_t)$, $\bar{a}_t = (1-\delta)a_t + \delta a'_t$, and $s_{t+1} \sim T_\rho(\cdot \mid s_t, \bar{a}_t)$.
        Estimate the policy gradient (where $G_t = \sum_{s=0}^{T} \gamma^s r_{t+s}$)
            $g = \frac{1-\delta}{|\mathcal{D}_k^{(n)}|} \sum_{\tau \in \mathcal{D}_k^{(n)}} \sum_t \gamma^t G_t^{(\tau)} \left[ \nabla_{\boldsymbol{y}} \log \pi_{\boldsymbol{y}}(a_t^{(\tau)} \mid s_t^{(\tau)}) \right]_{\boldsymbol{y} = \boldsymbol{y}_k^{(n)}}$
            $g' = \frac{\delta}{|\mathcal{D}_k^{(n)}|} \sum_{\tau \in \mathcal{D}_k^{(n)}} \sum_t \gamma^t G_t^{(\tau)} \left[ \nabla_w \log \pi_w(a_t'^{(\tau)} \mid s_t^{(\tau)}) \right]_{w = w_k^{(n)}}$
        $m \leftarrow \alpha m + (1-\alpha)\, g \odot g$ ; $C \leftarrow \mathrm{diag}\left(\sqrt{m + \epsilon}\right)$
        $\boldsymbol{y}_k^{(n+1)} \leftarrow \boldsymbol{y}_k^{(n)} + \eta C^{-1} g + \sqrt{2\eta}\sigma_k C^{-\frac{1}{2}} \xi$, where $\xi \sim \mathcal{N}(0, I)$
        $\bar{\boldsymbol{y}}_k \leftarrow (1-\beta)\, \bar{\boldsymbol{y}}_k + \beta \boldsymbol{y}_k^{(n+1)}$
        $m' \leftarrow \alpha m' + (1-\alpha)\, g' \odot g'$ ; $D \leftarrow \mathrm{diag}\left(\sqrt{m' + \epsilon}\right)$
        $w_k^{(n+1)} \leftarrow w_k^{(n)} - \eta D^{-1} g + \sqrt{2\eta}\sigma_k D^{-\frac{1}{2}} \xi'$, where $\xi' \sim \mathcal{N}(0, I)$
        $\bar{w}_k \leftarrow (1-\beta)\, \bar{w}_k + \beta w_k^{(n+1)}$
    end for
    $\boldsymbol{y}_{k+1} \leftarrow (1-\beta)\, \boldsymbol{y}_k + \beta \bar{\boldsymbol{y}}_k$
    $w_{k+1} \leftarrow (1-\beta)\, w_k + \beta \bar{w}_k$
end for
```

Algorithm 17: VPG with GAD (pre-conditioner = RMSProp) / Extra-Adam

```
Hyperparameters: see Table D.6
Initialize (randomly) policy parameters $\boldsymbol{y}_0$, $w_0$
for $k = 0, 1, 2, \ldots$ do
    Collect set of trajectories $\mathcal{D}_k = \{(\ldots, s_t^{(\tau)}, \bar{a}_t^{(\tau)}, r_t^{(\tau)}, \ldots)\}_\tau$ by running $\pi_{\boldsymbol{y}_k}$, and $\pi'_{w_k}$ in $\mathcal{M}$, i.e., $a_t \sim \pi_{\boldsymbol{y}_k}(s_t)$, $a'_t \sim \pi'_{w_k}(s_t)$, $\bar{a}_t = (1-\delta)a_t + \delta a'_t$, and $s_{t+1} \sim T_\rho(\cdot \mid s_t, \bar{a}_t)$.
    Estimate the policy gradient (where $G_t = \sum_{s=0}^{T} \gamma^s r_{t+s}$)
        $$g = \frac{1-\delta}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_t \gamma^t G_t^{(\tau)} \left[ \nabla_{\boldsymbol{y}} \log \pi_{\boldsymbol{y}}(a_t^{(\tau)} \mid s_t^{(\tau)}) \right]_{\boldsymbol{y}=\boldsymbol{y}_k}$$
        $$g' = \frac{\delta}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_t \gamma^t G_t^{(\tau)} \left[ \nabla_w \log \pi'_w({a'_t}^{(\tau)} \mid s_t^{(\tau)}) \right]_{w=w_k}$$
    GAD (pre-conditioner = RMSProp):
    $m \leftarrow \alpha m + (1-\alpha)\, g \odot g\, ;\ C \leftarrow \mathrm{diag}\left(\sqrt{m+\epsilon}\right)$
    $\boldsymbol{y}_{k+1} \leftarrow \boldsymbol{y}_k + \eta C^{-1} g$
    $m' \leftarrow \alpha m' + (1-\alpha)\, g' \odot g'\, ;\ D \leftarrow \mathrm{diag}\left(\sqrt{m'+\epsilon}\right)$
    $w_{k+1} \leftarrow w_k - \eta D^{-1} g'$
    Extra-Adam: use Algorithm 4 from [GBV+19].
end for
```



(a) $\delta = 0.1$



(b) $\delta = 0$

Figure D.19: Average performance (over 5 seeds) of Algorithm 16, and Algorithm 17 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$ and 0 (training on nominal environment $\rho_0 = 0.2$). The evaluation is performed without adversarial perturbations, on a range of environment parameters not encountered during training.

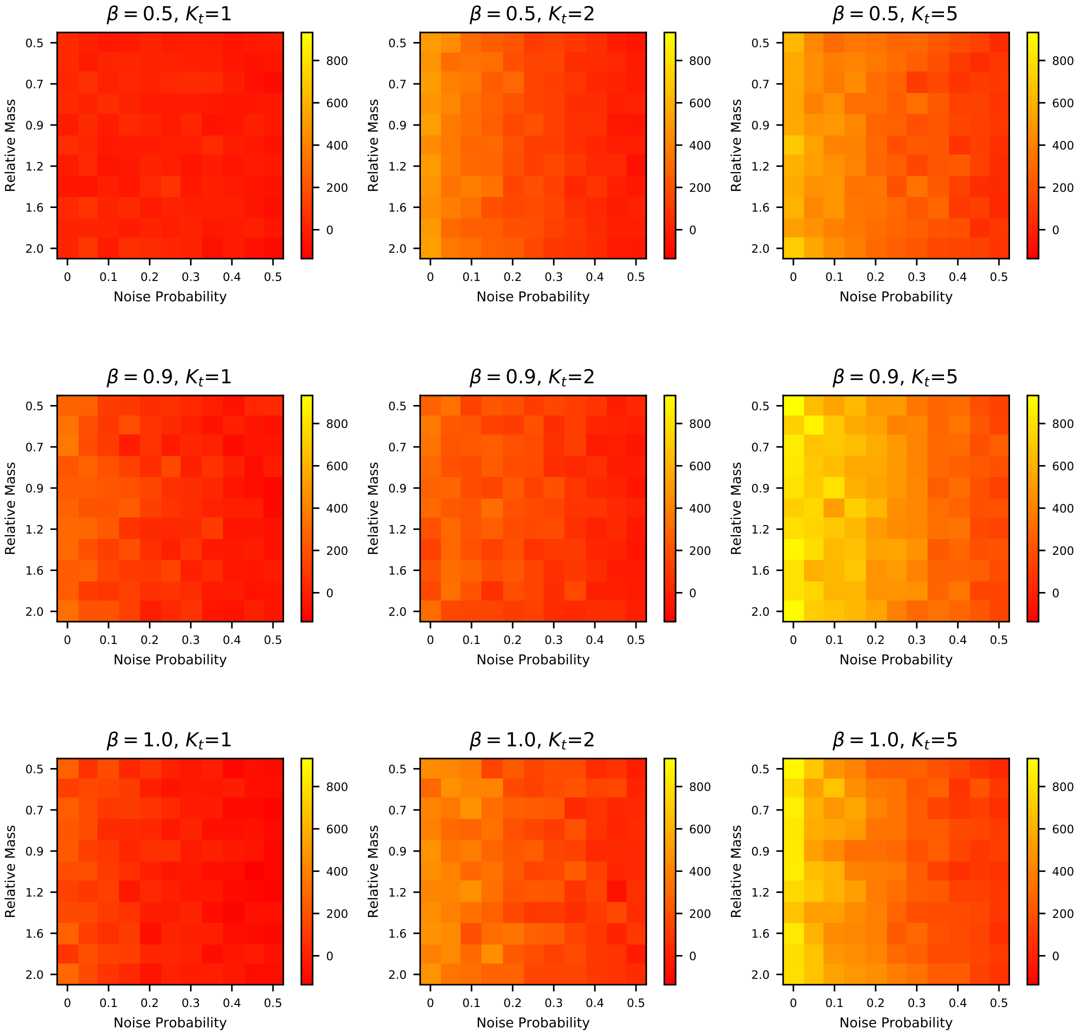


Figure D.20: Ablation study: Average performance (over 5 seeds) of MixedNE-LD (with different β, K_t) under the NR-MDP setting with $\delta = 0.1$ (training on Half-cheetah with relative mass 1). The evaluation is performed on a range of noise probability and mass values not encountered during training.

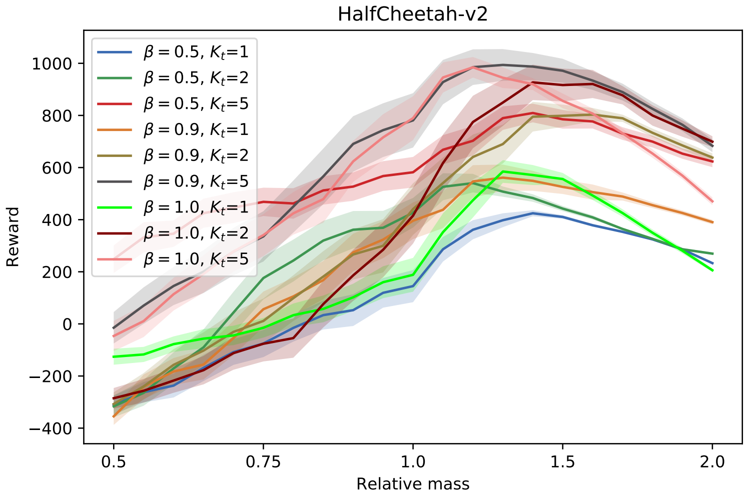


Figure D.21: Ablation study: Average performance (over 5 seeds) of MixedNE-LD (with different β, K_t) under the NR-MDP setting with $\delta = 0.1$ (training on Half-cheetah with relative mass 1). The evaluation is performed without adversarial perturbations, on a range of mass values not encountered during training.

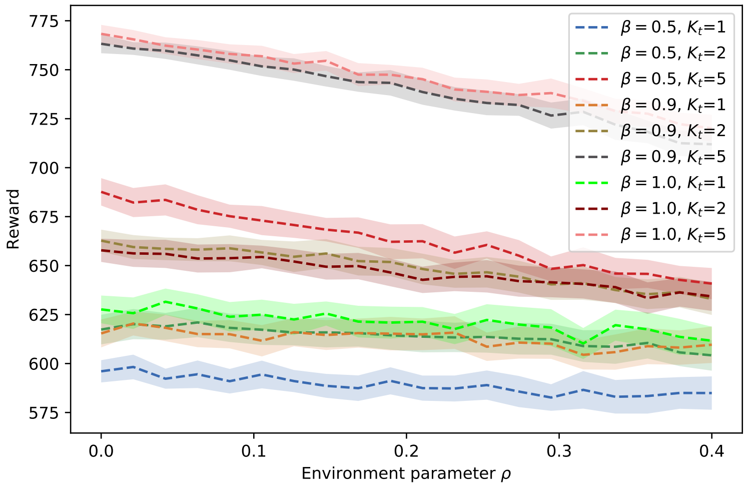


Figure D.22: Ablation study: Average performance (over 5 seeds) of MixedNE-LD (with different β, K_t) under the NR-MDP setting with $\delta = 0$ (training on nominal environment $\rho_0 = 0.2$). The evaluation is performed without adversarial perturbations, on a range of environment parameters not encountered during training.

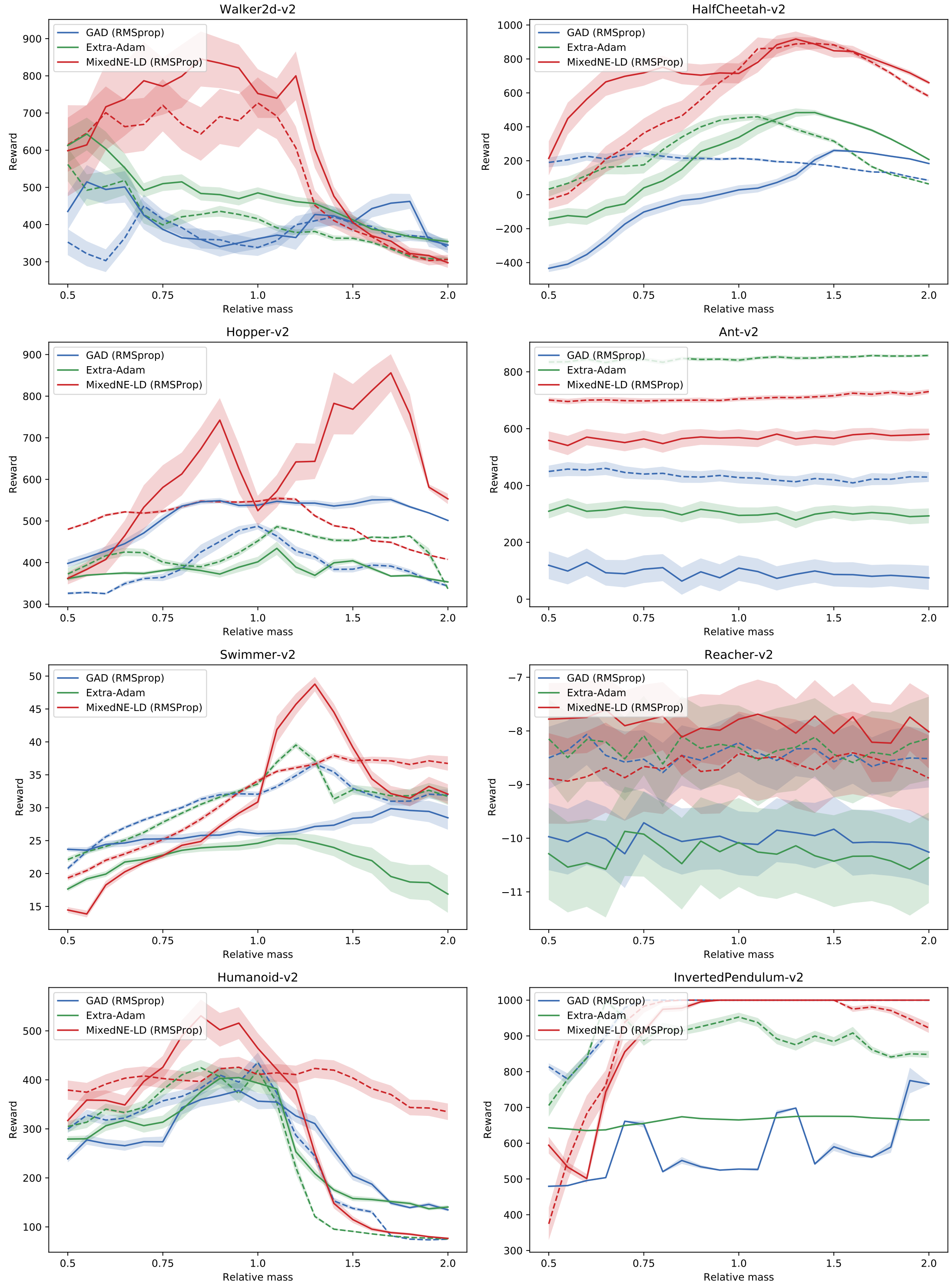


Figure D.23: Ablation study: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$ (solid lines) and $\delta = 0$ (dashed lines). The evaluation is performed without adversarial perturbations, on a range of mass values not encountered during training.

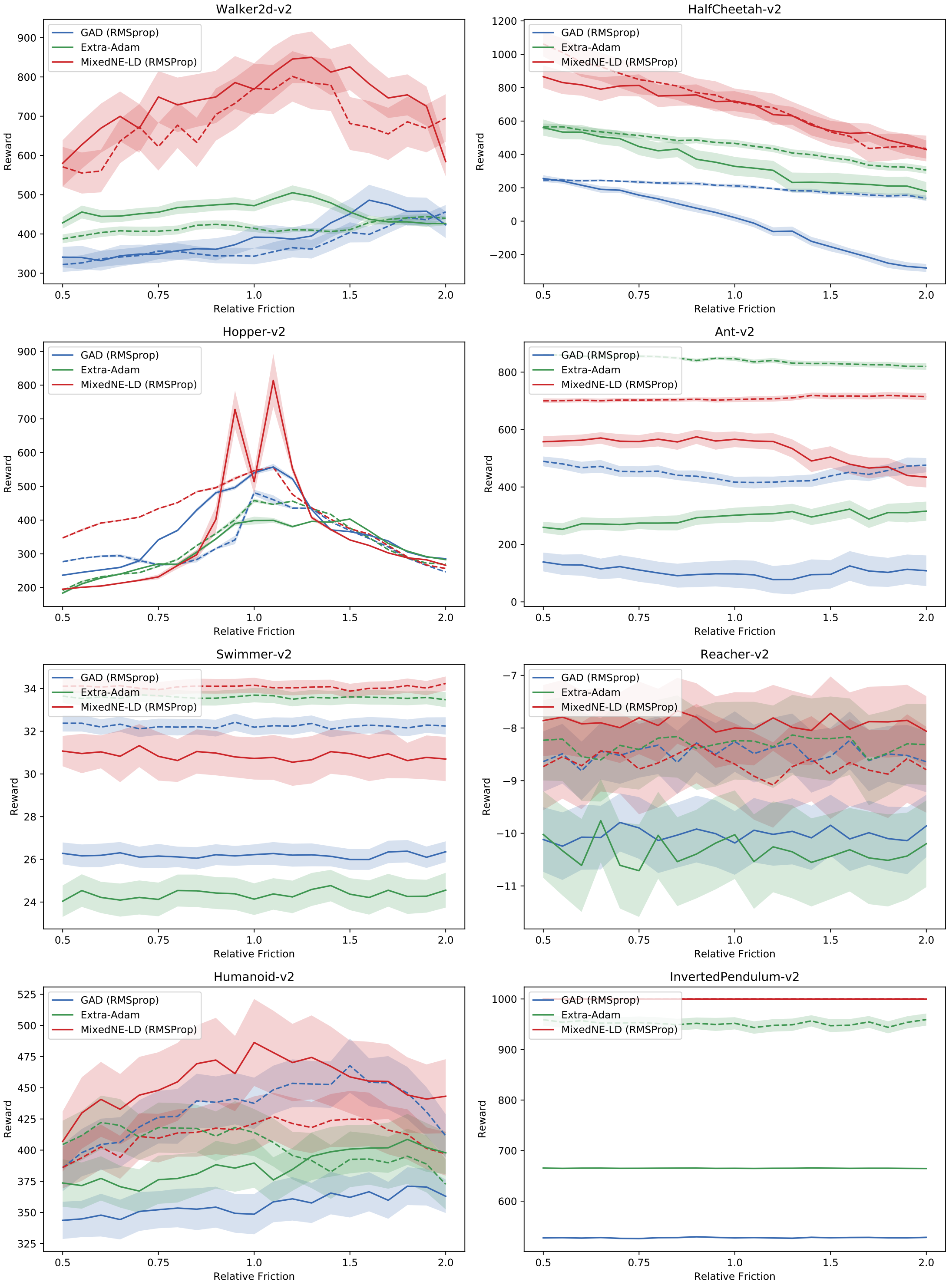


Figure D.24: Ablation study: Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$ (solid lines) and $\delta = 0$ (dashed lines). The evaluation is performed without adversarial perturbations, on a range of friction values not encountered during training.

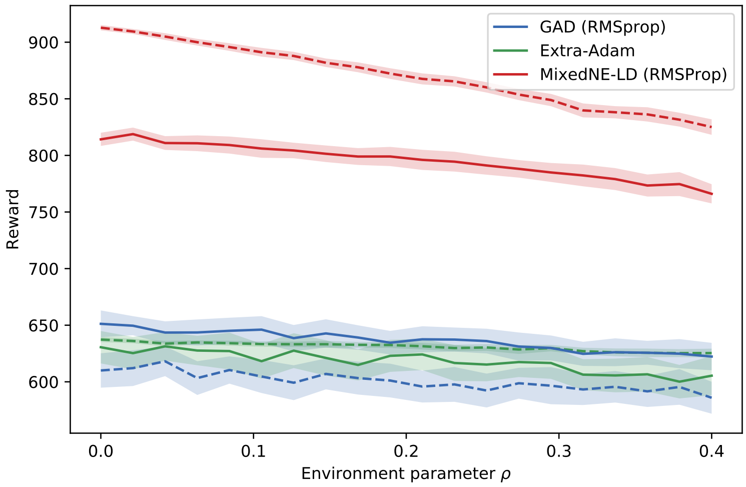


Figure D.25: Ablation study: Average performance (over 5 seeds) of different algorithms under the NR-MDP setting with $\delta = 0.1$ (solid lines) and $\delta = 0$ (dashed lines). The evaluation (after training on the nominal environment $\rho_0 = 0.2$) is performed without adversarial perturbations, on a range of environment parameters not encountered during training.

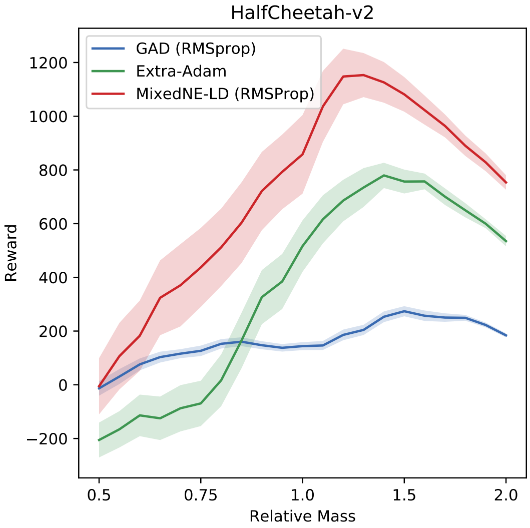


Figure D.26: $\delta = 0.1$

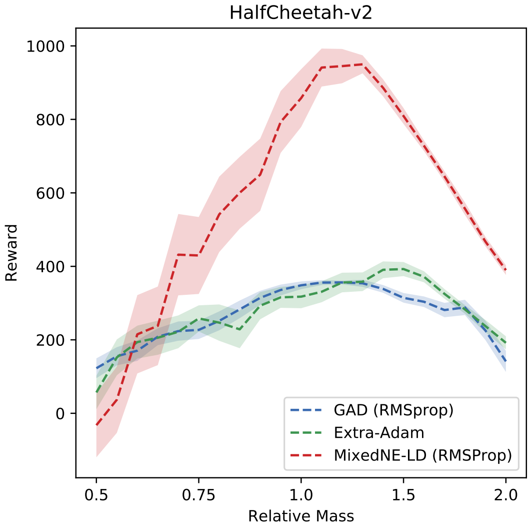


Figure D.27: $\delta = 0$

Figure D.28: HalfCheetah-v2 is trained over 2M steps. Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$ (solid lines) and $\delta = 0$ (dashed lines). The evaluation is performed without adversarial perturbations, on a range of mass values not encountered during training.

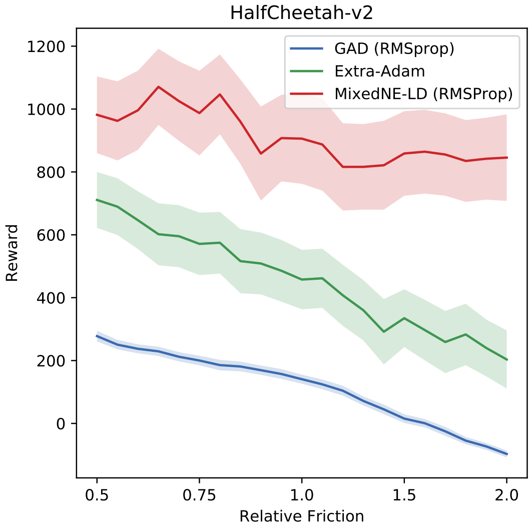


Figure D.29: $\delta = 0.1$

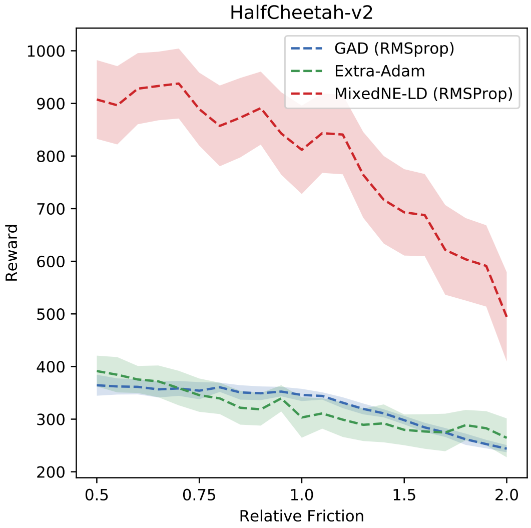


Figure D.30: $\delta = 0$

Figure D.31: HalfCheetah-v2 is trained over 2M steps. Average performance (over 5 seeds) of Algorithm 14, and Algorithm 15 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$ (solid lines) and $\delta = 0$ (dashed lines). The evaluation is performed without adversarial perturbations, on a range of friction values not encountered during training.

Bibliography

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[ACBG02] Peter Auer, Nicolo Cesa-Bianchi, and Claudio Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1):48–75, 2002.

[ADLH19] Leonard Adolphs, Hadi Daneshmand, Aurelien Lucchi, and Thomas Hofmann. Local saddle point optimization: A curvature exploitation approach. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 486–495, 2019.

[AHU58] Kenneth Joseph Arrow, Leonid Hurwicz, and Hirofumi Uzawa. *Studies in linear and non-linear programming*. Stanford University Press, 1958.

[AKW12] Sungjin Ahn, Anoop Korattikara, and Max Welling. Bayesian posterior sampling via stochastic gradient fisher scoring. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1771–1778, 2012.

[ALW19] Jacob Abernethy, Kevin A Lai, and Andre Wibisono. Last-iterate convergence rates for min-max optimization. *arXiv preprint arXiv:1906.02027*, 2019.

[AMLJG19] Waïss Azizian, Ioannis Mitliagkas, Simon Lacoste-Julien, and Gauthier Gidel. A tight and unified analysis of extragradient for a whole spectrum of differentiable games. *arXiv preprint arXiv:1906.05945*, 2019.

[BBF18] Sebastian Bervoets, Mario Bravo, and Mathieu Faure. Learning with minimal information in continuous games. https://arxiv.org/abs/1806.11506, 2018.

[BCL$^+$18] Sébastien Bubeck, Michael B Cohen, Yin Tat Lee, James R Lee, and Aleksander Madry. K-server via multiscale entropic regularization. In *Proceedings of the 50th annual ACM SIGACT symposium on theory of computing*, pages 3–16, 2018.

[BCP$^+$16] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[BD96] Odile Brandière and Marie Duflo. Les algorithmes stochastiques contournent-ils les pièges ? *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 32(3):395–427, 1996.

[Ben99] Michel Benaïm. Dynamics of stochastic approximation algorithms. In Jacques Azéma, Michel Émery, Michel Ledoux, and Marc Yor, editors, *Séminaire de Probabilités XXXIII*, volume 1709 of *Lecture Notes in Mathematics*, pages 1–68. Springer Berlin Heidelberg, 1999.

[BH95] Michel Benaïm and Morris W. Hirsch. Dynamics of Morse-Smale urn processes. *Ergodic Theory and Dynamical Systems*, 15(6):1005–1030, December 1995.

[BH96] Michel Benaïm and Morris W. Hirsch. Asymptotic pseudotrajectories and chain recurrent flows, with applications. *Journal of Dynamics and Differential Equations*, 8(1):141–176, 1996.

[BHS05] Michel Benaïm, Josef Hofbauer, and Sylvain Sorin. Stochastic approximations and differential inclusions. *SIAM Journal on Control and Optimization*, 44(1):328–348, 2005.

[BHS06] Michel Benaïm, Josef Hofbauer, and Sylvain Sorin. Stochastic approximations and differential inclusions, part II: Applications. *Mathematics of Operations Research*, 31(4):673–695, 2006.

[BLM18] Mario Bravo, David S. Leslie, and Panayotis Mertikopoulos. Bandit learning in concave N-person games. In *NeurIPS '18: Proceedings of the 32nd International Conference of Neural Information Processing Systems*, 2018.

[BM17] Mario Bravo and Panayotis Mertikopoulos. On the robustness of learning in games with stochastically perturbed payoff observations. *Games and Economic Behavior*, 103, John Nash Memorial issue:41–66, May 2017.

[BMP90] Albert Benveniste, Michel Métivier, and Pierre Priouret. *Adaptive Algorithms and Stochastic Approximations*. Springer, 1990.

[BNJ03] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[Bor18] Ali Borji. Pros and cons of gan evaluation measures. *arXiv preprint arXiv:1802.03446*, 2018.

[Bow75] Rufus Bowen. Omega limit sets of Axiom A diffeomorphisms. *Journal of Differential Equations*, 18:333–339, 1975.

[BRM$^+$18] David Balduzzi, Sebastien Racaniere, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. The mechanics of n-player differentiable games. In *International Conference on Machine Learning*, pages 354–363, 2018.

[BS18] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

[BT00] Dimitri P. Bertsekas and John N. Tsitsiklis. Gradient convergence in gradient methods with errors. *SIAM Journal on Optimization*, 10(3):627–642, 2000.

[BT03] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[Bub13a] Sebastien Bubeck. Orf523: Mirror descent, part i/ii, 2013.

[Bub13b] Sebastien Bubeck. Orf523: Mirror descent, part ii/ii, 2013.

[Bub13c] Sebastien Bubeck. Orf523: Mirror prox, 2013.

[Bur73] Donald Lyman Burkholder. Distribution function inequalities for martingales. *Annals of Probability*, 1(1):19–42, 1973.

[CBL06] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[CCH$^+$15] David E Carlson, Edo Collins, Ya-Ping Hsieh, Lawrence Carin, and Volkan Cevher. Preconditioned spectral descent for deep learning. In *Advances in Neural Information Processing Systems*, pages 2971–2979, 2015.

[CCS$^+$17] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. In *International Conference on Learning Representations*, 2017.

[CFG14] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International Conference on Machine Learning*, pages 1683–1691, 2014.

[CGFLJ19] Tatjana Chavdarova, Gauthier Gidel, François Fleuret, and Simon Lacoste-Julien. Reducing noise in GAN training with variance reduced extragradient. In *NeurIPS '19: Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019.

[CHC$^+$16] David Carlson, Ya-Ping Hsieh, Edo Collins, Lawrence Carin, and Volkan Cevher. Stochastic spectral descent for discrete graphical models. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):296–311, 2016.

[CHM17] Johanne Cohen, Amélie Héliou, and Panayotis Mertikopoulos. Learning with bandit feedback in potential games. In *NIPS '17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[CL07] Colin Christopher and Chengzhi Li. *Limit cycles of differential equations*. Springer Science & Business Media, 2007.

[Con78] Charles Cameron Conley. *Isolated Invariant Set and the Morse Index*. American Mathematical Society, Providence, RI, 1978.

[COO+18] Pratik Chaudhari, Adam Oberman, Stanley Osher, Stefano Soatto, and Guillaume Carlier. Deep relaxation: partial differential equations for optimizing deep neural networks. *Research in the Mathematical Sciences*, 5(3):30, Jun 2018.

[CYL+12] Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. In *Conference on Learning Theory*, pages 6–1, 2012.

[DDK11] Constantinos Daskalakis, Alan Deckelbaum, and Anthony Kim. Near-optimal no-regret algorithms for zero-sum games. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete algorithms*, pages 235–254. Society for Industrial and Applied Mathematics, 2011.

[DDK15] Constantinos Daskalakis, Alan Deckelbaum, and Anthony Kim. Near-optimal no-regret algorithms for zero-sum games. *Games and Economic Behavior*, 92:327–348, 2015.

[DEJM+20] Carles Domingo-Enrich, Samy Jelassi, Arthur Mensch, Grant Rotskoff, and Joan Bruna. A mean-field analysis of two-player zero-sum games. *arXiv preprint arXiv:2002.06277*, 2020.

[DFB+14] Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D Skeel, and Hartmut Neven. Bayesian sampling using stochastic gradient thermostats. In *Advances in neural information processing systems*, pages 3203–3211, 2014.

[DFT13] John C Doyle, Bruce A Francis, and Allen R Tannenbaum. *Feedback control theory*. Courier Corporation, 2013.

[DHK+17] Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. https://github.com/openai/baselines, 2017.

[DHS11] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(Jul):2121–2159, 2011.

[DISZ18] Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training GANs with optimism. In *International Conference on Learning Representations*, 2018.

[DMM18] Alain Durmus, Szymon Majewski, and Błażej Miasojedow. Analysis of langevin monte carlo via convex optimization. *arXiv preprint arXiv:1802.09188*, 2018.

[DP18] Constantinos Daskalakis and Ioannis Panageas. The limit points of (optimistic) gradient descent in min-max optimization. In *Advances in Neural Information Processing Systems*, pages 9236–9246, 2018.

[DP19] Constantinos Daskalakis and Ioannis Panageas. Last-iterate convergence: Zero-sum games and constrained min-max optimization. *Innovations in Theoretical Computer Science*, 2019.

[DPF14] Guido De Philippis and Alessio Figalli. The monge–ampère equation and its link to optimal transportation. *Bulletin of the American Mathematical Society*, 51(4):527–580, 2014.

[DR18] Gintare Karolina Dziugaite and Daniel Roy. Entropy-sgd optimizes the prior of a pac-bayes bound: Generalization properties of entropy-sgd and data-dependent priors. In *International Conference on Machine Learning*, pages 1376–1385, 2018.

[DSM+16] Alain Durmus, Umut Simsekli, Eric Moulines, Roland Badeau, and Gaël Richard. Stochastic gradient richardson-romberg markov chain monte carlo. In *Advances in Neural Information Processing Systems*, pages 2047–2055, 2016.

[EHHV+17] Marwa El Halabi, Ya-Ping Hsieh, Bang Vu, Quang Nguyen, and Volkan Cevher. General proximal gradient method: A case for non-euclidean norms. Technical report, 2017.

[FCR19] Tanner Fiez, Benjamin Chasnov, and Lillian J Ratliff. Convergence of learning dynamics in stackelberg games. *arXiv preprint arXiv:1906.01217*, 2019.

[FKG10] Bela A Frigyik, Amol Kapila, and Maya R Gupta. Introduction to the dirichlet distribution and related processes. *Department of Electrical Engineering, University of Washignton, UWEETR-2010-0006*, 2010.

[FVGP19] Lampros Flokas, Emmanouil Vasileios Vlatakis-Gkaragkounis, and Georgios Piliouras. Poincaré recurrence, cycles and spurious equilibria in gradient-descent-ascent for non-convex non-concave zero-sum games. In *NeurIPS '19: Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019.

[FvHM18] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.

[GAA$^+$17] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[GBV$^+$19] Gauthier Gidel, Hugo Berard, Gaëtan Vignoud, Pascal Vincent, and Simon Lacoste-Julien. A variational inequality perspective on generative adversarial networks. In *ICLR '19: Proceedings of the 2019 International Conference on Learning Representations*, 2019.

[GHJY15] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points — Online stochastic gradient for tensor decomposition. In *COLT '15: Proceedings of the 28th Annual Conference on Learning Theory*, 2015.

[GHP$^+$19] Gauthier Gidel, Reyhane Askari Hemmat, Mohammad Pezeshki, Rémi Le Priol, Gabriel Huang, Simon Lacoste-Julien, and Ioannis Mitliagkas. Negative momentum for improved game dynamics. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1802–1811, 2019.

[Gib02] J Willard Gibbs. *Elementary principles in statistical mechanics*. Yale University Press, 1902.

[Gli52] Irving L Glicksberg. A further generalization of the kakutani fixed point theorem, with application to nash equilibrium points. *Proceedings of the American Mathematical Society*, 3(1):170–174, 1952.

[GPAM$^+$14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[Gra11] Robert M Gray. *Entropy and information theory*. Springer Science & Business Media, 2011.

[Hal13] Paul R Halmos. *Measure theory*, volume 18. Springer, 2013.

[HC18] Ya-ping Hsieh and Volkan Cevher. Dimension-free information concentration via exp-concavity. In *Algorithmic Learning Theory*, pages 451–469, 2018.

[HH80] P. Hall and C. C. Heyde. *Martingale Limit Theory and Its Application*. Probability and Mathematical Statistics. Academic Press, New York, 1980.

[HIMM19] Yu-Guan Hsieh, Franck Iutzeler, Jérôme Malick, and Panayotis Mertikopoulos. On the convergence of single-call stochastic extra-gradient methods. In *NeurIPS '19: Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 6936–6946, 2019.

[HKM$^+$18] Ya-Ping Hsieh, Yu-Chun Kao, Rabeeh Karimi Mahabadi, Alp Yurtsever, Anastasios Kyrillidis, and Volkan Cevher. A non-euclidean gradient descent framework for non-convex matrix factorization. *IEEE Transactions on Signal Processing*, 66(22):5917–5926, 2018.

[HKRC18] Ya-Ping Hsieh, Ali Kavis, Paul Rolland, and Volkan Cevher. Mirrored langevin dynamics. In *Advances in Neural Information Processing Systems*, pages 2878–2887, 2018.

[HLC19] Ya-Ping Hsieh, Chen Liu, and Volkan Cevher. Finding mixed nash equilibria of generative adversarial networks. In *International Conference on Machine Learning*, pages 2810–2819, 2019.

[HMC20] Ya-Ping Hsieh, Panayotis Mertikopoulos, and Volkan Cevher. The limits of min-max optimization algorithms: convergence to spurious non-critical sets. *arXiv preprint arXiv:2006.09065*, 2020.

[HRU$^+$17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[JN11] Anatoli Juditsky and Arkadi Nemirovski. First order methods for nonsmooth convex large-scale optimization, ii: utilizing problems structure. *Optimization for Machine Learning*, pages 149–183, 2011.

[JNJ19] Chi Jin, Praneeth Netrapalli, and Michael I Jordan. What is local optimality in nonconvex-nonconcave minimax optimization? *arXiv preprint arXiv:1902.00618*, 2019.

[JNT11] Anatoli Juditsky, Arkadi Semen Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58, 2011.

[Jor19] Michael I Jordan. Artificial intelligence—the revolution hasn't happened yet. *Harvard Data Science Review*, 1(1), 2019.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KB15] Diederik P Kingma and Jimmy Ba. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, volume 5, 2015.

[KC78] Harold J. Kushner and D. S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer, 1978.

[KHH$^+$20] Parameswaran Kamalaruban, Yu-Ting Huang, Ya-Ping Hsieh, Paul Rolland, Cheng Shi, and Volkan Cevher. Robust reinforcement learning via adversarial training with langevin dynamics. *arXiv preprint arXiv:2002.06063*, 2020.

[KHSC18] Ehsan Asadi Kangarshahi, Ya-Ping Hsieh, Mehmet Fatih Sahin, and Volkan Cevher. Let's be honest: An optimal no-regret framework for zero-sum games. In *International Conference on Machine Learning*, pages 2488–2496, 2018.

[Kor76] G. M. Korpelevich. The extragradient method for finding saddle points and other problems. *Èkonom. i Mat. Metody*, 12:747–756, 1976.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[KW52] Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466, 1952.

[KY97] Harold J. Kushner and G. G. Yin. *Stochastic approximation algorithms and applications*. Springer-Verlag, New York, NY, 1997.

[LCCC16] Chunyuan Li, Changyou Chen, David E Carlson, and Lawrence Carin. Preconditioned stochastic gradient langevin dynamics for deep neural networks. In *AAAI*, 2016.

[Lee03] John M. Lee. *Introduction to Smooth Manifolds*. Number 218 in Graduate Texts in Mathematics. Springer-Verlag, New York, NY, 2003.

[Let20] Alistair Letcher. On the impossibility of global convergence in multi-loss optimization. *arXiv preprint arXiv:2005.12649*, 2020.

[LHP$^+$15] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[LHZC15] Yen-Huan Li, Ya-Ping Hsieh, Nissim Zerbib, and Volkan Cevher. A geometric view on constrained m-estimators. *arXiv preprint arXiv:1506.08163*, 2015.

[Lju77] Lennart Ljung. Analysis of recursive stochastic algorithms. *IEEE Trans. Autom. Control*, 22(4):551–575, August 1977.

[Lju86] Lennart Ljung. *System Identification Theory for the User*. Prentice Hall, Englewood Cliffs, NJ, 1986.

[LKM$^+$18] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In *Advances in neural information processing systems*, 2018.

[LLC$^+$19] Sijia Liu, Songtao Lu, Xiangyi Chen, Yao Feng, Kaidi Xu, Abdullah Al-Dujaili, Minyi Hong, and Una-May Obelilly. Min-max optimization without gradients: Convergence and applications to adversarial ml. *arXiv preprint arXiv:1909.13806*, 2019.

[LMR$^+$19] Mingrui Liu, Youssef Mroueh, Jerret Ross, Wei Zhang, Xiaodong Cui, Payel Das, and Tianbao Yang. Towards better understanding of adaptive gradient algorithms in generative adversarial nets. *arXiv* *preprint arXiv:1912.11940*, 2019.

[LS16] Shiwei Lan and Babak Shahbaba. Sampling constrained probability distributions using spherical augmentation. In *Algorithmic Advances in Riemannian Geometry and Applications*, pages 25–71. Springer, 2016.

[LS19] Tengyuan Liang and James Stokes. Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 907–915, 2019.

[LZS16] Chang Liu, Jun Zhu, and Yang Song. Stochastic gradient geodesic mcmc methods. In *Advances in Neural Information Processing Systems*, pages 3009–3017, 2016.

[Man83] Benoit B Mandelbrot. *The fractal geometry of nature*, volume 173. WH freeman New York, 1983.

[MCF15] Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient mcmc. In *Advances in Neural Information Processing Systems*, pages 2917–2925, 2015.

[MD05] Jun Morimoto and Kenji Doya. Robust reinforcement learning. *Neural computation*, 17(2):335–359, 2005.

[MJS19] Eric V Mazumdar, Michael I Jordan, and S Shankar Sastry. On finding local nash equilibria (and only local nash equilibria) in zero-sum games. *arXiv preprint arXiv:1901.00838*, 2019.

[MLZ$^+$19] Panayotis Mertikopoulos, Bruno Lecouat, Houssam Zenati, Chuan-Sheng Foo, Vijay Chandrasekhar, and Georgios Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. In *ICLR '19: Proceedings of the 2019 International Conference on Learning Representations*, 2019.

[MMS$^+$18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. 2018.

[MNG17] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of gans. In *Advances in Neural Information Processing Systems*, pages 1825–1835, 2017.

[MOP19] Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. *arXiv preprint arXiv:1901.08511*, 2019.

[MPP18] Panayotis Mertikopoulos, Christos H. Papadimitriou, and Georgios Piliouras. Cycles in adversarial regularized learning. In *SODA '18: Proceedings of the 29th annual ACM-SIAM Symposium on Discrete Algorithms*, 2018.

[MRS20] Eric Mazumdar, Lillian J Ratliff, and S Shankar Sastry. On gradient-based learning in continuous games. *SIAM Journal on Mathematics of Data Science*, 2(1):103–131, 2020.

[MT20] Yura Malitsky and Matthew K Tam. A forward-backward splitting method for monotone inclusions without cocoercivity. *SIAM Journal on Optimization*, 30(2):1451–1472, 2020.

[Mye99] Roger B Myerson. Nash equilibrium and the history of economic theory. *Journal of Economic Literature*, 37(3):1067–1082, 1999.

[MZ19] Panayotis Mertikopoulos and Zhengyuan Zhou. Learning in games with continuous action sets and unknown payoff functions. *Mathematical Programming*, 173(1-2):465–507, January 2019.

[Nas50] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the USA*, 36:48–49, 1950.

[Nem04] Arkadi Nemirovski. Prox-method with rate of convergence o $(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[Nes04] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course.* Number 87 in Applied Optimization. Kluwer Academic Publishers, 2004.

[Nes09] Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259, 2009.

[NFBB$^+$16] Ashkan Norouzi-Fard, Abbas Bazzi, Ilija Bogunovic, Marwa El Halabi, Ya-Ping Hsieh, and Volkan

Cevher. An efficient streaming algorithm for the submodular cover problem. In *Advances in Neural Information Processing Systems*, pages 4493–4501, 2016.

[NI19] Roi Naveiro and David Ríos Insua. Gradient methods for solving stackelberg games. In *International Conference on Algorithmic DecisionTheory*, pages 126–140. Springer, 2019.

[NJLS09] Arkadi Semen Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[NK17] Vaishnavh Nagarajan and J Zico Kolter. Gradient descent gan optimization is locally stable. In *Advances in neural information processing systems*, pages 5585–5595, 2017.

[NSH$^+$19] Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D Lee, and Meisam Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. In *Advances in Neural Information Processing Systems*, pages 14905–14916, 2019.

[NY83] AS Nemirovsky and DB Yudin. Problem complexity and method efficiency in optimization. 1983.

[OLY$^+$16] Gergely Odor, Yen-Huan Li, Alp Yurtsever, Ya-Ping Hsieh, Quoc Tran-Dinh, Marwa El Halabi, and Volkan Cevher. Frank-wolfe works for non-lipschitz continuous gradient objectives: scalable poisson phase retrieval. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6230–6234. Ieee, 2016.

[PDSG17] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2817–2826. JMLR. org, 2017.

[PDZC20] Wei Peng, Yu-Hong Dai, Hui Zhang, and Lizhi Cheng. Training gans with centripetal acceleration. *Optimization Methods and Software*, pages 1–19, 2020.

[Pea94] Barak A Pearlmutter. Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160, 1994.

[Pem90] Robin Pemantle. Nonconvergence to unstable points in urn models and stochastic aproximations. *Annals of Probability*, 18(2):698–712, April 1990.

[PL12] Steven Perkins and David S. Leslie. Asynchronous stochastic approximation with differential inclusions. *Stochastic Systems*, 2(2):409–446, 2012.

[PML17] Steven Perkins, Panayotis Mertikopoulos, and David S. Leslie. Mixed-strategy learning with continuous action sets. *IEEE Trans. Autom. Control*, 62(1):379–384, January 2017.

[Pop80] Leonid Denisovich Popov. A modification of the Arrow–Hurwicz method for search of saddle points. *Mathematical Notes of the Academy of Sciences of the USSR*, 28(5):845–848, 1980.

[PT13] Sam Patterson and Yee Whye Teh. Stochastic gradient riemannian langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems*, pages 3102–3110, 2013.

[RCJ19] Arvind Raghunathan, Anoop Cherian, and Devesh Jha. Game theoretic optimization via gradient-based nikaido-isoda function. In *International Conference on Machine Learning*, pages 5291–5300, 2019.

[RM51] Herbert Robbins and Sutton Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[RMC15] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[Roc15] Ralph Tyrell Rockafellar. *Convex analysis*. Princeton university press, 2015.

[RRT17] Maxim Raginsky, Alexander Rakhlin, and Matus Telgarsky. Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. In *Conference on Learning Theory*, pages 1674–1703, 2017.

[RS13a] Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. In *Conference on Learning Theory*, pages 993–1019, 2013.

[RS13b] Alexander Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pages 3066–3074, 2013.

[SA19] Florian Schäfer and Anima Anandkumar. Competitive gradient descent. In *Advances in Neural Information Processing Systems*, pages 7623–7633, 2019.

[SBCR16] Umut Simsekli, Roland Badeau, Taylan Cemgil, and Gaël Richard. Stochastic quasi-newton langevin monte carlo. In *International Conference on Machine Learning*, pages 642–651, 2016.

[SMSM00] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[Spa92] James C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Autom. Control*, 37(3):332–341, March 1992.

[TEM19] Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. *arXiv preprint arXiv:1901.09184*, 2019.

[TET12] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5026–5033. IEEE, 2012.

[TTV16] Yee Whye Teh, Alexandre H Thiery, and Sebastian J Vollmer. Consistency and fluctuations for stochastic gradient langevin dynamics. *The Journal of Machine Learning Research*, 17(1):193–225, 2016.

[UO30] George E Uhlenbeck and Leonard S Ornstein. On the theory of the brownian motion. *Physical review*, 36(5):823, 1930.

[VAHC20] Maria-Luiza Vladarean, Ahmet Alacaoglu, Ya-Ping Hsieh, and Volkan Cevher. Conditional gradient methods for stochastically constrained convex minimization. In *International Conference on Machine Learning*, 2020.

[Vil03] Cédric Villani. *Topics in optimal transportation.* Number 58. American Mathematical Soc., 2003.

[Vil08] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[vN28] John von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100:295–320, 1928. Translated by S. Bargmann as “On the Theory of Games of Strategy” in A. Tucker and R. D. Luce, editors, *Contributions to the Theory of Games IV*, volume 40 of *Annals of Mathematics Studies*, pages 13-42, 1957, Princeton University Press, Princeton.

[vNM44] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, 1944.

[Wig03] Stephen Wiggins. *Introduction to applied nonlinear dynamical systems and chaos*, volume 2. Springer Science & Business Media, 2003.

[WT11] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

[YHC15] Alp Yurtsever, Ya-Ping Hsieh, and Volkan Cevher. Scalable convex methods for phase retrieval. In *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 381–384. IEEE, 2015.

[YSX$^+$18] Abhay Yadav, Sohil Shah, Zheng Xu, David Jacobs, and Tom Goldstein. Stabilizing adversarial nets with prediction methods. In *ICLR ’18: Proceedings of the 2018 International Conference on Learning Representations*, 2018.

[YSZ$^+$15] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[ZLC17] Yuchen Zhang, Percy Liang, and Moses Charikar. A hitting time analysis of stochastic gradient langevin dynamics. In *Conference on Learning Theory*, pages 1980–2022, 2017.

[ZLHC16] Nissim Zerbib, Yen-Huan Li, Ya-Ping Hsieh, and Volkan Cevher. Estimation error of the constrained lasso. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 433–438. IEEE, 2016.

[ZY19] Guojun Zhang and Yaoliang Yu. Convergence of gradient methods on bilinear zero-sum games, 2019.

Ya-Ping Hsieh

Curriculum Vitae

"Theory is the first term in the Taylor series of practice" - Thomas Cover

Education

2015–2020 **Ph.D. in Electrical Engineering (advisor: Volkan Cevher)**.
École Polytechnique Fédérale de Lausanne

2019 Winter **Visiting student (advisor: Jason Lee)**.
Princeton University

2010–2012 **M.S. in Communication Engineering and Signal Processing**.
National Taiwan University

2006–2010 **B.S. in Electrical Engineering and Computer Science**.
National Taiwan University

Employment Experience

2015–Present **Ph.D. student**, École Polytechnique Fédérale de Lausanne, Switzerland.
Research Interests:
- Theory for deep learning.
- Sampling algorithms.
- Online learning.
- Concentration of measure, optimal transport, Riemannian geometry.

2012–2014 **Research Assistant**, Academia Sinica, Taiwan.
- Quantum information and quantum statistics.
- Information theory in nano-communication.

Awards

2020 ETH-FDS Postdoctoral Fellow

2015 Best Paper Award, Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) 2015.

Publication

2020 M.-L. Vladarean, A. Alacaoglu, Y.-P. Hsieh, and V. Cevher, “Conditional gradient methods for stochastically constrained convex minimization,” in *37th International Conference on Machine Learning (ICML), 2020.*

☎ *+41 78 932 04 38* • ✉ *ya-ping.hsieh@epfl.ch*

- Y.-P. Hsieh, P. Mertikopoulos, and V. Cevher, "The limits of min-max optimization algorithms: convergence to spurious non-critical sets," in *arXiv preprint arXiv:2006.09065, 2020.*

- P. Kamalaruban, Y.-T. Huang, Y.-P. Hsieh, P. Rolland, C. Shi, and V. Cevher, "Robust reinforcement learning via adversarial training with langevin dynamics," in *arXiv preprint arXiv:2002.06063, 2020.*

2019 Y.-P. Hsieh, C. Liu, and V. Cevher, "Finding mixed nash equilibria of generative adversarial networks," in *36th International Conference on Machine Learning (ICML), 2019.*

2018 Y.-P. Hsieh, C. Liu, and V. Cevher, "Finding mixed nash equilibria of generative adversarial networks," in *Advances in Neural Information (NeurIPS) Processing Systems, Smooth Games Optimization and Machine Learning Workshop*, 2018. (**Oral**)

- Y.-P. Hsieh, A. Kavis, P. Rolland, and V. Cevher, "Mirrored langevin dynamics," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. (**Spotlight**)

- (*equal contribution)E. A. Kangarshahi*, Y.-P. Hsieh*, M. F. Sahin, and V. Cevher, "Let's be honest: An optimal no-regret framework for zero-sum games," in *35th International Conference on Machine Learning (ICML), 2018.* (**Long talk**)

- Y.-P. Hsieh and V. Cevher, "Dimension-free information concentration via exp-concavity," in *Algorithmic Learning Theory (ALT)*, pp. 451–469, 2018.

- Y.-P. Hsieh, Y.-C. Kao, R. Karimi Mahabadi, Y. Alp, A. Kyrillidis, and V. Cevher, "A Non-Euclidean Gradient Descent Framework for Non-Convex Matrix Factorization," *accepted to IEEE Transactions on Signal Processing*, 2018.

2017 M. El Halabi, Y.-P. Hsieh, B. Vu, Q. Nguyen, and V. Cevher, "General proximal gradient method: A case for non-euclidean norms," tech. rep., 2017.

2016 A. Norouzi, B. Abbas, I. Bogunovic, M. El-Halabi, Y.-P. Hsieh, and V. Cevher, "Efficient algorithm for streaming submodular cover," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

- N. Zerbib, Y.-H. Li, Y.-P. Hsieh, and V. Cevher, "Estimation error of the lasso," in *54th Annual Allerton Conference on Communication, Control, and Computing, University of Illinois at Urbana-Champaign, Urbana*, 2016.

- D. Carlson, Y.-P. Hsieh, E. Collins, L. Carin, and V. Cevher, "Stochastic spectral descent for discrete graphical models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 296–311, 2016.

- G. Odor, Y.-H. Li, A. Yurtsever, Y.-P. Hsieh, Q. Tran-Dinh, M. El Halabi, and V. Cevher, “Frank-wolfe works for non-lipschitz continuous gradient objectives: Scalable poisson phase retrieval,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6230–6234, IEEE, 2016.

2015 D. E. Carlson, E. Collins, Y.-P. Hsieh, L. Carin, and V. Cevher, “Preconditioned spectral descent for deep learning,” in *Advances in Neural Information Processing Systems (NIPS)*, pp. 2971–2979, 2015.

- A. Yurtsever, Y.-P. Hsieh, and V. Cevher, “Scalable convex methods for phase retrieval,” in *IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP),* , pp. 381–384, IEEE, 2015. (**Best Paper Award**)

- Y.-H. Li, Y.-P. Hsieh, N. Zerbib, and V. Cevher, “A geometric view on constrained m-estimators,” *arXiv preprint arXiv:1506.08163*, 2015.

2013 Y.-P. Hsieh and P.-C. Yeh, “Mathematical foundations for information theory in diffusion-based molecular communications,” *arXiv preprint arXiv:1311.4431*, 2013.

- Y.-P. Hsieh, Y.-C. Lee, P.-J. Shih, P.-C. Yeh, and K.-C. Chen, “On the asynchronous information embedding for event-driven systems in molecular communications,” *Nano Communication Networks*, vol. 4, no. 1, pp. 2–13, 2013.

2012 Y.-P. Hsieh, P.-J. Shih, Y.-C. Lee, P.-C. Yeh, and K.-C. Chen, “An asynchronous communication scheme for molecular communication,” in *2012 IEEE International Conference on Communications (ICC)*, pp. 6177–6182, IEEE, 2012.

Invited Talk

2018 **Mirrored Langevin Dynamics**, *ML theory & methodology meeting, EPFL*, Switzerland.

- **Mirrored Langevin Dynamics**, *Alan Turing Institute*, England.

Languages

Chinese Mothertongue
English Fluent

Programming Skills

Python (Pytorch, TensorFlow)
Matlab