

Understanding Representation Learning for Deep Reinforcement Learning



Charline Le Lan
Jesus College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Trinity 2023

To my parents.

Acknowledgements

This thesis is the result of an immense support I received from many brilliant and caring people who made my time as a PhD student a truly outstanding journey. I am deeply grateful to my supervisors, mentors, collaborators, friends and family for their constant guidance and encouragement over the years. Without them, there is no denying that this thesis would not have been possible.

First, I would like to thank Marc G. Bellemare and Yee Whye Teh for advising me throughout this PhD.

Marc is an outstanding supervisor and I am forever grateful to him for his invaluable support. Four years ago I knew very little about reinforcement learning. Still Marc generously gave me the wonderful opportunity to work with him and patiently guided me through every step of this DPhil. He taught me how to frame suitable research questions, how to read and write papers and how to do good science. When working on research, I often found myself wondering "What would Marc do here?". Marc is also an extraordinary researcher. I have always admired his vision, ideas and analysis. While I often came to our meetings with confusions and issues, I always left with a clear understanding and direction, thrilled to dig into a new result. He also has this ability for linking concepts that are usually considered distinct which was a constant source of inspiration during my DPhil. I was also truly impressed by the care and time he devotes to his students and his dedication to training the best possible researchers. On top of this, Marc truly is a great person. He created many exciting research opportunities for me, I whole-heartily thank him for this. Marc, it was an honor to be your student. Thank you for believing in me and for your kindness! Our memories of doing research together and your passion will stay with me for a long time.

Yee Whye Teh enabled me to enter the field of machine learning research when I first reached out to him four years ago. Despite not having any prior research experience, he took a chance on me and provided me with the tremendous opportunity to work in his group at Oxford. This had a significant impact on my life. I am also grateful for his kindness and experience and for supporting me in achieving my career goals. Thank you for giving me the research freedom and time to find my own direction, research agenda and style.

I am extremely grateful to have had the opportunity to collaborate with so many wonderful collaborators: Rishabh Agarwal, Stephen Tu, Pablo Samuel Castro, Laurent Dinh, Mark Rowland, Will Dabney, Jesse Farebrother, Joshua Greaves, Fabian Pedregosa, Anna Harutyunyan, Ross Goroshin, Adam Oberman, Emile Mathieu and Ryota Tomioka.

Rishabh has been a fantastic colleague and friend over the course of my PhD. It was always fascinating to chat about research with him. I am impressed by the breath of his knowledge and thankful for all he kindly taught me about deep reinforcement learning.

Stephen taught me so much about learning theory, matrices and probability while we were working together. He was always eager to engage with problems I was stuck on and provided me with the tools I needed to develop my research. I feel indebted for all the time he generously offered to me.

Pablo has been an amazing mentor, especially at the beginning of my PhD. He offered me the exciting opportunity to host me at Google Brain in 2019 and taught me a lot while we worked together on our bisimulation project which became my first first-author paper. I also admire Pablo as a person, he is very generous and kind.

Laurent has been very supportive at the beginning of my PhD journey, even when I had no clear research path at times. He shared with me his ideas, taught me how to organize a paper and how to give talks. I am deeply thankful for his expertise, compassion and humanity.

It was a joy working with Mark and Will on the last paper of this DPhil. They brought new perspectives to me both in terms of ideas and ways of doing research and I am very appreciative of that.

By working with Joshua Greaves, Jesse Farebrother and Pablo Samuel Castro, my programming skills have significantly improved, I am very grateful to them.

I had the good fortune of visiting several institutions during my DPhil, Google Brain, MILA and DeepMind.

I was very fortunate to meet Adrien Ali Taiga, Max Schwarzer, Linda Petrini, Harley Wiltzer, Nathan U. Rahn, Pierluca D'Oro, Jacob Buckman and Johan Obando Céron. Thank you for your friendship and many stimulating discussions.

My time at Google has significantly influenced the direction of my research. I feel so privileged to have been part of such a wonderful research group. Many thanks to all the individuals who enriched my time there, created exciting opportunities for me and always made me feel welcome, among them Hugo Larochelle, Olivier Pietquin, Danny Tarlow, Robert Dadashi, Liam Fedus, Vincent Dumoulin, Marlos Machado, Matthieu Geist, Dale Schuurman, Hanie Sedghi, Damien Vincent, Nino Vieillard, Leonard Hussenot, Johan Ferret, Chris Dann, Mathieu Blondel, Marcin Andrychowicz and Nicolas Le Roux. I would also like to thank my fellow interns,

John D. Martin, Ahmed Touati, Saurabh Kumar, Carles Gelada, Erin Grant, Eleni Triantafillou and Khimya Khetarpal, who made my internship even more enjoyable.

I would like to thank the numerous individuals I had the opportunity to interact with at DeepMind, including Yunhao Tang, Bernardo Avila Pires, Daniel Guo, David Abel, Shantanu Thakoor, Remi Munos, Diana Borsa, Mo Azar, Georg Ostrovski and Zeyu Zheng. I also had many fruitful discussions with Theophane Weber, Bilal Piot, Evgenii Nikishin, David Parkes, Chris Grimm, Andre Barreto, Angelos Filos, Hado Van Hasselt, Tom Schaul, Ian Gemp, Abbas Abdolmaleki and Akhil Bagaria.

I would also like to acknowledge Shimon Whiteson and the students from the WhiRL lab for offering valuable discussions and letting me selectively join their reading group.

I extend my heartfelt thanks to Martha White and Patrick Rebeschini for examining my thesis.

I also appreciated the support of Francois-Xavier Briol and Mark Girolami who mentored me during my master's thesis at Imperial College London.

Thank you to Clare Lyle, Sephora Madjiheurem and Laura Toni with whom I enjoyed many stimulating conversations about representation learning and reinforcement learning.

My colleagues and friends at the Department of Statistics have greatly enhanced my life in Oxford. In particular, I appreciated the support of Emilien Dupont, Jean-Francois Ton, Jin Xu, Sheheryar Zaidi, Adam Kosiorek, Hyunjik Kim, Giuseppe Di Benedetto, Qinyi Zhang, Frauke Harms and Xiaoyu Lu who were all awesome office mates. Many thanks also go to Emile Mathieu, Adam Foster, Chris Maddison, Bobby He, Dominic Richards, Bryn Elesedy, Adam Golinski, Michael Hutchinson, Tim Rudner, Joost van Amersfoort, Aidan Gomez, Cong Lu, Faaiz Taufiq, Tom Rainforth, Bradley Gram-Hansen, Xenia Miscouridou, Yuan Zhou, Tomas Vaskevicius, Deborah Sulem, Serte Donderwinkel, Ian Letter, Amartya Sanyal and Luisa Zintgraf.

Thanks also go to Mark Brooke, Robin Wang and Ella Butcherine for being the best flatmates and making me feel at home in the beautiful city of Oxford.

I have been very lucky to have had the great company of several friends both in and outside Oxford. I would like to express my special thanks to my peers from prepa. I am indebted to Alexandre Poka for his many years of friendship, Léo Aparisi de Lannoy who in some sense inspired me to pursue this DPhil, Benoît Pit-Claudiel, Hortense Jamet, Julie Zhang and Cedric Oppé. I am also thankful to Benjamin

Levai for keeping up with me and for all the joyful conversations. Together with my childhood friends Marie-Astrid and Charles-Edouard Sevilla, we have experienced so much and I look forward to many more memorable trips in the years to come.

I want to particularly thank Marietta Almasy for giving me the opportunity to pursue my hobby outside of research. Marietta has always been a source of inspiration and her passion and success have motivated me over the years. Thank you also to Cecile Taleux, Aurore Voisin, France Mentre, Isabelle Nobile, Anna Brunel and Franck Prazan for creating a vibrant community around this hobby.

Finally, my deepest gratitude goes towards my family. I am thankful to my brother, Nicolas, for his encouragements and unwavering help, even in difficult times. My parents, Karine and Jean-Claude, have been my pillar of support. Thank you for all the sacrifices you made for me and for letting me pursue interests of my own. You were always here to cheer me up when I was doubting myself and celebrate in moments of success. Words cannot express how thankful I am for your unconditional love.

Thank you.

Abstract

Representation learning is essential to practical success of reinforcement learning. Through a state representation, an agent can describe its environment to efficiently explore the state space, generalize to new states and perform credit assignment from delayed feedback. These representations may be state abstractions, hand-engineered or fixed features or implied by a neural network. In this thesis, we investigate several desirable theoretical properties of state representations and, using this categorization, design novel principled RL algorithms aiming at learning these state representations at scale through deep learning.

First, we consider state abstractions induced by behavioral metrics and their generalization properties. We show that supporting the continuity of the value function is central to generalization in reinforcement learning. Together with this formalization, we provide an empirical evaluation comparing various metrics and demonstrating the importance of the choice of a neighborhood in RL algorithms.

Then, we draw on statistical learning theory to characterize what it means for arbitrary state features to generalize in RL. We introduce a new notion called effective dimension of a representation that drives the generalization to unseen states and demonstrate its usefulness for value-based deep reinforcement learning in Atari games.

The third contribution of this dissertation is a scalable algorithm to learn a state representation from a very large number of auxiliary tasks through deep learning. It is a stochastic gradient descent method to learn the principal components of a target matrix by means of a neural network from a handful of entries.

Finally, the last part presents our findings on how the state representation in reinforcement learning influences the quality of an agent's predictions but is also shaped by these predictions. We provide a formal mathematical model for studying this phenomenon and show how these theoretical results can be leveraged to improve the quality of the learning process.

Contents

List of Figures	xv
List of Tables	xxi
1 Introduction	1
1.1 Motivation	1
1.2 Thesis Outline	2
1.3 Publications	5
2 Background and Related Work	7
2.1 Reinforcement Learning	7
2.2 Representation Learning	9
2.2.1 Feature Engineering in Classical Reinforcement Learning . .	9
2.2.2 You Are What You Predict: Representation Learning in Deep Reinforcement Learning	11
2.2.3 Representations Matter	13
3 Metrics and Continuity in Reinforcement Learning	17
3.1 Introduction	20
3.2 Overview	21
3.3 Background	22
3.3.1 Metrics, Topologies, and Continuity	24
3.3.2 Prior Metrics and Abstractions	25
3.4 Continuity Relationships	28
3.5 Taxonomy of Metrics	29
3.5.1 Continuity: Prior Metrics	29
3.5.2 Value-Based Metrics	31
3.5.3 Categorizing Metrics, Continuity and Complexity	32
3.6 Empirical Evaluation	33
3.6.1 Generalizing the Value Function V^*	34
3.6.2 Generalizing the Q-function Q^*	35
3.6.3 Approximate Value Iteration	35
3.7 Discussion	36

3.8	Broader Impact	37
3.A	Proofs for Section 3.4	38
3.B	Proofs for Section 3.5	47
3.C	Formal Definition of Bisimulation Metrics	55
3.D	Additional Empirical Evaluations	56
4	On the Generalization of Representations in Reinforcement Learning	59
4.1	Introduction	62
4.2	Background	64
4.2.1	Statistical Learning Theory	65
4.2.2	The Successor Representation	66
4.3	Characterizing Excess Risk	67
4.3.1	Illustrative Examples	70
4.4	Generalization for the Successor Representation	71
4.4.1	Approximation Error: $\ P_{\Phi}^{\perp}V^{\pi}\ _{S,2}^2$	71
4.4.2	Effect of Transition Structure	73
4.4.3	Analysis of the One-dimensional Torus	75
4.5	Experiments	76
4.5.1	Comparing State Representations	76
4.5.2	Deep Reinforcement Learning	77
4.6	Conclusion	79
4.A	Proofs for Section 4.3	80
4.B	Proofs for Section 4.4	86
4.C	Empirical Evaluation: Additional Details	89
4.C.1	Graphical Structures	89
4.C.2	Full Atari Results	93
4.D	Societal Impact	98
5	A Novel Stochastic Gradient Descent Algorithm for Learning Principal Subspaces	101
5.1	Introduction	104
5.2	Background	105
5.2.1	Problem Statement	105
5.3	PCA from Samples	107
5.3.1	An Improved Gradient Estimate	109
5.3.2	Estimate of the Weight Vector $W_{\Phi,t}^*$	110
5.3.3	Algorithm Based on LISSA	112
5.4	Related Work	114
5.5	Experiments	115

5.5.1	Synthetic Matrices	116
5.5.2	MNIST Dataset	118
5.5.3	Learning the Successor Measure	119
5.6	Discussion & Conclusion	121
5.A	Proofs for Section 5.2	122
5.B	Proofs for Section 5.3	123
5.C	Additional Experimental Results	126
5.C.1	Synthetic Matrices	126
5.C.2	MNIST	127
5.C.3	Puddle World	127
6	Bootstrapped Representations in Reinforcement Learning	131
6.1	Introduction	134
6.2	Background	136
6.2.1	Auxiliary Tasks	137
6.2.2	Monte Carlo Representations	138
6.2.3	Temporal Difference Learning with a Deep Network	139
6.3	Bootstrapped Representations	140
6.4	Representations for Policy Evaluation	144
6.4.1	TD and Monte Carlo Need Different Cumulants	146
6.4.2	A Deeper Analysis of Random Cumulants	147
6.5	Empirical Analysis	148
6.5.1	Synthetic Matrices	148
6.5.2	Effectiveness of Random Cumulants	149
6.5.3	Offline Pre-training	150
6.6	Related Work	152
6.7	Conclusion	153
6.A	Additional Empirical Results	154
6.A.1	Additional Details for Subsection 6.5.1	154
6.A.2	Additional Details for Subsection 6.5.2	154
6.A.3	Additional Details for Subsection 6.5.3	155
6.B	Proofs for Section 6.2	157
6.C	Proofs for Section 6.3	158
6.D	Proofs for Section 6.4	164
6.E	Proofs for Subsection 6.4.1	167
6.F	Proofs for Subsection 6.4.2	170
6.F.1	Notations	170
6.F.2	Approximate Matrix Decompositions	171
6.F.3	Analysis	173

7 Discussion	177
7.1 Conclusion	177
7.2 Future Directions	178
7.2.1 Further Theoretical Analysis of Representation Learning Schemes	178
7.2.2 Benchmarks	180
7.2.3 Pre-training Representations and Reincarnating Reinforce- ment Learning	181
Bibliography	183

List of Figures

2.1	The Markov decision process model [Sutton and Barto, 2018]. . . .	7
2.2	A deep RL architecture reproduced from Bellemare et al. [2023] . . .	11
3.1	A simple five-state MDP (top) with the neighbourhoods induced by three metrics: an identity metric which isolates each state (d_1); a metric which captures behavioral proximity (d_2); and a metric which is not able to distinguish states (d_3). The yellow circles represent ϵ -balls in the corresponding metric spaces. The bottom row indicates the V^* values for each state.	20
3.2	Errors when approximating the optimal value function (left) and optimal Q-function (center) via nearest-neighbours and errors when performing value iteration on aggregated states (right). Curves for e^\sim and $e^{\sim \iota ax}$ are covering each other on all of the plots. Averaged over 100 Garnet MDPs with 200 states and 5 actions, with 50 independent runs for each (to account for subsampling differences). Confidence intervals were very tiny due to the large number of runs so were not included.	34
3.3	Four Rooms domain with a single goal state in green (left). Optimal values for each cell (right).	56
3.4	The top row illustrates the distances from the top-left cell to every other cell (note the color scales are shifted for each metric for easier differentiation between states). The bottom row displays $d(s, t) - V^*(s) - V^*(t) $, where s is the top-left cell, illustrating how tight an upper bound the metrics yield on the difference in optimal values.	57
3.5	State clusters produced by the different metrics when targeting 11 aggregate states. There is no color correlation across metrics.	57
4.1	A deep RL architecture seen as a deep representation ϕ and a value prediction $\hat{V}_{\phi, w}$	62

4.2	Singular values of the successor representation Ψ^π , in decreasing order and for different graphical structures. Note that the fully connected and star graphs' spectra overlap (top left). Effective dimension of the representation $\Phi_k = F_k$ (top right). Median empirical excess risk over 10 runs, with 95% CIs as shaded regions, and theoretical excess risk, respectively, for the open room, torus, and fully connected graphs (bottom left and right).	73
4.3	The Four Rooms domain (left). Median empirical excess risk (middle) and effective dimension (right) as a function of approximation error for the top k left singular vectors of the SR, random features, the Krylov basis and the bisimulation metric matrix in the Four rooms domain.	76
4.4	Interquartile mean (IQM) [Agarwal et al., 2021b] for the effective dimension, normalized by the batch size used $N = 2^{15}$ (left). Interquartile mean (IQM) for human-normalized scores over the course of training across 60 Atari games (right). IQM measures the mean on the middle 50% of the data points combined across all runs and games. These statistics are over 5 independent runs and shading gives 95% stratified bootstrap confidence intervals based on Rliable [Agarwal et al., 2021b].	77
4.5	Effective dimension, normalized by the batch size $N = 2^{15}$ and performance of IQN and IQN with feature regularization L_ϕ on 17 Atari games in the offline RL setting.	78
4.6	Different graphical structures with $S = 5$ states from left to right, Star, Chain, Torus1d, Disconnected, Fullyconnected (top). Two-dimensional graphical structures with $S = 9$ states: from left to right, Openroom and Torus2d (bottom).	89
4.7	Approximation error $\ P_{F_k} V^\pi\ $ given a one-hot, all-ones and Gaussian reward vector and for MDPs with different graphical structures (left). Median empirical excess risk $\mathcal{E}(V_{F_k, \hat{w}})$ given a one-hot, all-ones and Gaussian reward vector (middle). Theoretical excess risk for a representation $\Phi_k = F_k$ and a one-hot, all-ones and Gaussian reward vector (right). The median is over 5 random seeds and shading gives 95% confidence intervals.	92
4.8	Sweeping over various values of α when adding the auxiliary loss \mathcal{L}_ϕ to IQN.	93
4.9	Average estimate (darker color) of the effective dimension normalized by the batch size used $N = 2^{15}$ on DQN(Nature), DQN(Adam), Rainbow, IQN and M-IQN on all 60 Atari games computed using 5 independent runs. Individual runs are shown with a lighter color.	94

4.10	Per-game learning curves of IQN and IQN with feature regularization L_ϕ on 17 Atari games in the offline RL setting.	95
4.11	Per-game effective dimension normalized by the batch size $N = 2^{15}$ of IQN and IQN with feature regularization L_ϕ on 17 Atari games in the offline RL setting, using 5 independent runs. Individual runs are shown with a lighter color.	96
4.12	Per-game rank of IQN and IQN with feature regularization L_ϕ computed with a batch size $N = 2^{15}$ on 17 Atari games in the offline RL setting, using 5 independent runs. Individual runs are shown with a lighter color.	97
4.13	Interquartile mean (IQM) [Agarwal et al., 2021b] for the rank of representations induced by IQN and IQN with feature regularization L_ϕ computed with a batch size $N = 2^{15}$ on 17 Atari games in the offline setting.	98
5.1	Subspace distance over the course of training LISSA for different dimensions (left, $L = 25, J = M = N = 5$) and for different total number of samples per update (right, $d = 10$) on synthetic matrices with a spectrum decaying linearly and exponentially, averaged over 30 seeds. Shaded areas represent estimates of 95% confidence intervals.	116
5.2	Subspace distance ($d = 10$) after 10^6 training steps according to the method used to estimate the loss gradient. Here, the x axis represents the total number of row samples L from the Φ matrix with $J = M = N$ ($L = 2J + 2M + N$ for the Danskin methods, $J + M + N$ for the naive method). Shaded areas represent estimates of 95% confidence intervals. Note that because we are sampling with replacement, the gradient estimate for $L = 250$ still differs from the gradient given in Lemma 12. (The naive method diverges for very small values of L).	117
5.3	Training curves for LISSA on MNIST ($d = 16$) that updates only a subset of pixels at a time (left). *: see main text. Reconstruction on MNIST test images (right). First row show samples from test images. Second are images reconstructed from the true principal components of Ψ and third row are images reconstructed from the principal components learnt by Danskin-LISSA ($N = 64$). Reconstruction MSE errors for true components and Danskin-LISSA are 21.46 and 21.53 respectively.	118

5.4	The Puddle World domain [Sutton, 1995], with the shaded area indicating regions where the agent moves slowly (left). In our experiments, each grid cell is associated with a column of the implied data matrix. Subspace distance as a function of the dimension d after 10^8 gradient steps for three methods: Danskin-LISSA, Explicit, and the Large Batch baseline (right).	120
5.5	Subspace distance after 10^6 training steps of the LISSA algorithm for different κ_0	126
5.6	Subspace distance over the course of training LISSA for different dimensions on synthetic matrices with a spectrum decaying linearly and exponentially, averaged over 30 seeds. The total number of samples used is 50. Shaded areas represent 95% confidence intervals.	127
5.7	First 10 principal components of the successor measure of the Puddle World domain.	128
6.1	In deep RL, we see the penultimate layer of the network as the representation ϕ which is linearly transformed into a value prediction $\hat{V}_{\phi,w}$ and auxiliary predictions $\Psi(x)$ by bootstrapping methods. . . .	134
6.2	A simple 3-state MDP (left). Five subspaces, each represented by a circle, spanned by Φ during the last training steps of gradient descent on $\mathcal{L}_{\text{aux}}^{\text{TD}}$ for $d = 2$ (right).	142
6.3	MC (left) and TD (right) approximation errors as a function of the misalignment of the top left and right singular vector of the SR induced by greedifying the policy. Trained with $\mathcal{L}_{\text{aux}}^{\text{MC}}, \mathcal{L}_{\text{aux}}^{\text{TD}}, G = I, d = 1$ on a 4-state room.	145
6.4	Subspace distance between Φ and the top- d left singular vectors of the SR on the left (resp. and a top- d P^π -invariant subspace in the middle over the course of training $\mathcal{L}_{\text{aux}}^{\text{TD}}, \mathcal{L}_{\text{aux}}^{\text{MC}}$ and $\mathcal{L}_{\text{aux}}^{\text{res}}$ for 10^5 steps, averaged over 30 seeds ($d = 3$). MDPs with real diagonalisable (left, middle) and symmetric (right) transition matrices are randomly generated. Shaded areas represent 95% confidence intervals.	149
6.5	Subspace distance after 5×10^5 training steps and averaged over 30 seeds ($d = 5$) between Φ learnt with $\mathcal{L}_{\text{aux}}^{\text{MC}}$ and the top left singular vectors of the SR (left) and between Φ learnt with $\mathcal{L}_{\text{aux}}^{\text{TD}}$ and the top invariant subspaces of the SR (right) for different random cumulants, on the Four Rooms domain. Shaded areas represent estimates of 95% confidence intervals	150
6.6	Comparing effects of offline pre-training on the Four Rooms (left) and sparse Mountain Car (right) domains for different cumulant generation methods. Results are averages over three seeds.	151

6.7	Monte Carlo and TD approximation errors after $5 \cdot 10^5$ training steps on the learning rules $\mathcal{L}_{\text{aux}}^{\text{MC}}$ (on the left column) and $\mathcal{L}_{\text{aux}}^{\text{TD}}$ (on the right column) in the Four Rooms domain for different distributions of cumulant, averaged over 30 seeds, for $d = 5$. Shaded areas represent estimates of 95% confidence intervals.	154
6.8	Example for ExactSVD of the learned cumulants (first two rows) and value functions (last two rows) during offline pre-training in Four Rooms under the uniform random policy.	156
6.9	Example for Normal of the learned cumulants (first two rows) and value functions (last two rows) during offline pre-training in Four Rooms under the uniform random policy.	156
6.10	Example for CCR of the learned cumulants (first two rows) and value functions (last two rows) during offline pre-training in Four Rooms under the uniform random policy.	157
6.11	Example for RNI of the learned cumulants (first two rows) and value functions (last two rows) during offline pre-training in Four Rooms under the uniform random policy.	157
6.12	Example for ExactSVD of the learned cumulants (first two rows) and value functions (last two rows) during offline pre-training in sparse Mountain Car under the uniform random policy.	158
6.13	Example for Normal of the learned cumulants (first two rows) and value functions (last two rows) during offline pre-training in sparse Mountain Car under the uniform random policy.	158
6.14	Example for CCR of the learned cumulants (first two rows) and value functions (last two rows) during offline pre-training in sparse Mountain Car under the uniform random policy.	159
6.15	Example for RNI of the learned cumulants (first two rows) and value functions (last two rows) during offline pre-training in sparse Mountain Car under the uniform random policy.	159

List of Tables

3.1	Different types of state abstractions.	26
3.2	Categorization of state metrics, their continuity implications, and their complexity (when known). The notation $\{y\}^{\mathcal{S}}$ denotes any function $h : \mathcal{S} \rightarrow Y$ that is constant, $Y^{\mathcal{S}}$ refers to all functions $h : \mathcal{S} \rightarrow Y$. $\mathcal{B}(Y^{\mathcal{S}})$ (resp. $\mathcal{B}_L(Y^{\mathcal{S}})$) is a bounded (resp. locally bounded) function $h : \mathcal{S} \rightarrow Y$. “-” denotes an absence of LC, UC, ULC and LLC. In the complexity column, δ is the desired accuracy.	32
3.3	RL functions with their respective domains and ranges.	38
6.1	Different types of representation loss and their induced representations. The supervised targets $\Psi \in \mathbb{R}^{S \times T}$ are $(I - \gamma P^\pi)^{-1}G$. SVD(M) denotes the top- d left singular vectors of M, INV(M) the top- d invariant subspace of M and $\Sigma_d \in \mathbb{R}^{d \times d}$ the diagonal matrix with the top- d singular values of $(I - \gamma P^\pi)^{-1}$ on its diagonal.	144

1

Introduction

1.1 Motivation

The idea of interaction is central to intelligence and all theories for learning. In reinforcement learning (RL), an agent interacts with its environment by taking actions and receives a real-valued reward as a form of delayed feedback. The goal is to turn data into decisions to maximize this numerical reward signal. Unlike supervised learning which is concerned with learning from a dataset of labeled examples, reinforcement learning fundamentally aims at learning to act from data.

Reinforcement learning achieved a number of notable achievements such as playing games [Silver et al., 2016], flying stratospheric balloons [Bellemare et al., 2020], designing hardware chips [Mirhoseini et al., 2021], discovering matrix multiplication algorithms [Fawzi et al., 2022] and finetuning large language models [Christiano et al., 2017, Ouyang et al., 2022, OpenAI, 2023]. These recent successes can be attributed to deep reinforcement learning, that is the combination of reinforcement learning algorithms with deep neural networks. In deep reinforcement learning, the network learns the mapping from perceptual inputs such as raw pixels to an output vector encoding each action and needs to figure out how those low-level inputs are related. These algorithms help represent inputs in a way that captures

the relevant information needed for the agent to make good decisions, a process called representation learning.

Understanding how the choice of a representation affects the performance of deep RL agents, summarised by the sum of discounted rewards they receive, remains poorly understood. Representation learning in reinforcement learning is quite different from representation learning in supervised learning. In RL, an agent interacts with the environment in a temporal fashion. Hence, an advantage is that there is structure built into the decision making system that we can actually learn from. Images that occur in succession are more related than images that are far apart which relates to the problem of decisions: we should take the same decisions in similar situations. However, unlike supervised learning, in the control setting, the agent faces a succession of value prediction problems making representation learning more difficult.

In this thesis, we analyse what makes a good representation for reinforcement learning. In particular, we provide evidence of the need to take advantage of the structure of the interactions between the agent and its environment into a compact representation. More generally, our work unifies theoretical reinforcement learning with practical deep-learning-based algorithms and also provides the ground for principled deep reinforcement learning agents.

1.2 Thesis Outline

The research question behind this thesis is

How can we choose and learn a state representation to improve the quality of the learning process and its resulting solution in reinforcement learning?

This thesis answers this question by the following statement.

Thesis statement.

By leveraging insights from topology, statistical learning and control theory, we identify several theoretical properties for useful state representations leading to novel, efficient and principled reinforcement learning algorithms.

To support this thesis, we consider three desiderata according to which we evaluate representations for reinforcement learning. A representation should be easy to learn, cheaply generalize to newly encountered states while making accurate predictions about the value function. We study the characteristics of state representations under this lens through four contributions.

Chapter 3 studies state abstractions induced by behavioral metrics in which similar states are assigned similar predictions. State similarity metrics can support the continuity of RL functions to various degrees and induce different kinds of topologies on the state space. The main insight of this chapter is that generalizing well within a neighborhood requires having a representation that supports the continuity of the function of interest, as summarised by Table 3.2, and a topology as coarse as possible enabling a cheap generalization to new states as shown by Theorem 2. We also provide results comparing the computational cost of these metrics. Relying on our taxonomy, we show that metrics frequently found in the literature are not appropriate for algorithms that convert representations into values. Following this observation, we present new metrics to address this gap. We also provide empirical evidence supporting our taxonomy of metrics and showing the benefit of the state abstractions introduced to generalize values within a neighborhood.

Chapter 4 investigates how the choice of a representation affects the generalization of value functions in an algorithmic context. Theorem 6 provides a generalization bound for Monte Carlo value function estimation with fixed features. We demonstrate that our bound is useful by applying it to the special case of the successor representation for various environment structures. We find that a quantity that we call effective dimension of a representation informs its generalization capacity. Finally, we show that our theory makes useful predictions about which representations are desirable on the Arcade Learning Environment [Bellemare et al., 2013]. Our experiments highlight a strong correlation between the effective dimension of the representations implied by deep RL agents and their performance

in the online setting. We exploit this tight connection to design a new auxiliary loss presenting encouraging improvements in the offline deep RL setting.

Chapter 5 tackles learning a representation at scale from a possibly infinite number of predictions. It is motivated by a commonly held belief that the more an agent learns about the world additionally to learning a value function, the better its representation and resulting performance. We propose a gradient-based algorithm which applies beyond the setting of reinforcement learning and recovers the principal components of a possibly infinite dimensional data matrix by means of a neural network from a small number of entries. It consists in expressing a per-task weight vector implicitly rather than in memory and constructing an estimate of a loss function which minimizer is the desired principal subspace. Empirically, we demonstrate on tabular and continuous domains that the representation parameterized by a neural network effectively converges towards to the desired principal components of the matrix of interest.

Chapter 6 addresses the problem of designing a set of cumulants given a fixed computational budget. To answer this question, we first rely on theoretical tools from the study of dynamical systems to analyse the representations learnt by training auxiliary tasks. We also formalize mathematically why a collection of cumulants spanning the whole state space leads to rich representations, as has been suggested by Sutton et al. [2011] and show that learning the same representation from a smaller set of cumulants requires them to depend on the dynamics of the environment. We demonstrate empirically that our theoretical results make useful predictions about what happens in deep RL and which representations are useful in the control setting as verified in experiments in Subsection 6.5.3.

Together these contributions advance the study of representation learning in reinforcement learning by providing the tools to analyse the features learnt by state-of-the art RL algorithms, moving ahead with a formalization of why they are helpful in the learning process of the value function and concluding with the derivation of new auxiliary-task based algorithms.

1.3 Publications

Most chapters of this integrated thesis correspond to papers published in conference proceedings [Le Lan et al., 2021, 2022, 2023a] or presented at a workshop [Le Lan et al., 2023b]. We detail the contributions of each author at the end of each corresponding chapter.

- Charline Le Lan, Marc G. Bellemare, Pablo Samuel Castro. Metrics and continuity in reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021
- Charline Le Lan, Stephen Tu, Adam Oberman, Rishabh Agarwal, Marc G. Bellemare. On the generalization of representations in reinforcement learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2022
- Charline Le Lan, Joshua Greaves, Jesse Farebrother, Mark Rowland, Fabian Pedregosa, Rishabh Agarwal, Marc G. Bellemare. A novel stochastic gradient descent algorithm for learning principal subspaces. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2023
- Charline Le Lan, Stephen Tu, Mark Rowland, Anna Harutyunyan, Rishabh Agarwal, Marc G. Bellemare, Will Dabney. Bootstrapped representations in reinforcement learning. In *Reincarnating RL Workshop at ICLR 2023*

For completeness, we list in reverse chronological order below other publications conducted during the time of this DPhil. Some of them are discussed as related work [Le Lan and Agarwal, 2023, Farebrother et al., 2023, Tang et al., 2023] while others are omitted from this thesis [Le Lan and Dinh, 2021, Hutchinson, Le Lan, Zaidi et al., 2021, Mathieu et al., 2019]. * denotes joint first authorship.

- Charline Le Lan, Rishabh Agarwal. Revisiting bisimulation: a sampling-based state similarity pseudo-metric. In *Submission at the International Conference on Learning Representation, Tiny paper track*, 2023

- Jesse Farebrother*, Joshua Greaves*, Rishabh Agarwal, Charline Le Lan, Ross Ghoroshin, Pablo Samuel Castro, Marc G. Bellemare. Proto-value networks: scaling representation learning with auxiliary tasks. In *Proceedings of the International Conference on Learning Representations, 2023*
- Yunhao Tang, Zhaohan Daniel Guo, Pierre Harvey Richemond, Bernardo Avila Pires, Yash Chandak, Remi Munos, Mark Rowland, Mohammad Gheshlaghi Azar, Charline Le Lan, Clare Lyle, Andras Gyorgy, Shantanu Thakoor, Will Dabney, Bilal Piot, Daniele Calandriello, Michal Valko. Understanding self-predictive learning for reinforcement learning. In *Submission at the International Conference on Machine Learning, 2023*
- Charline Le Lan, Laurent Dinh. Perfect density models cannot guarantee anomaly detection. In *Entropy, 2021*. Also *Entropic Award* at the *I Can't Believe It's Not Better! Workshop at NeurIPS, 2020*
- Michael Hutchinson*, Charline Le Lan*, Sheheryar Zaidi*, Emilien Dupont, Yee Whye Teh, Hyunjik Kim. Lietransformer: equivariant self-attention for lie groups. In *Proceedings of the International Conference on Machine Learning, 2021*
- Emile Mathieu, Charline Le Lan, Chris Maddison, Ryota Tomioka, Yee Whye Teh. Continuous hierarchical representations with poincaré variational auto-encoders. In *Advances in Neural Information Processing Systems, 2019*

2

Background and Related Work

We start by reviewing two areas central to this work: reinforcement learning and representation learning. This chapter describes some background material that will be necessary for the understanding of the following chapters. For convenience, some of these concepts are also recalled at the beginning of each subsequent chapter, where necessary.

2.1 Reinforcement Learning

In reinforcement learning, an *agent* interacts with an *environment* modeled as a discrete-time *Markov Decision Process* (MDP). Formally, an MDP is a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$ [Puterman, 1994] with state space \mathcal{S} , set of actions \mathcal{A} , transition kernel $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{P}(\mathcal{S})$, deterministic reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow [-R_{\max}, R_{\max}]$, and *discount factor* $\gamma \in [0, 1)$ which reflects that it is preferable to

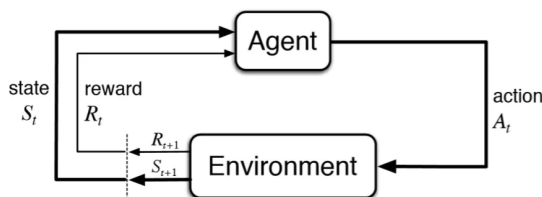


Figure 2.1: The Markov decision process model [Sutton and Barto, 2018].

receive rewards sooner than later. Figure 2.1 illustrates this model.

The agent starts in an initial state $S_0 \sim \xi_0$ where $\xi_0 \in \mathcal{P}(\mathcal{S})$ is a probability distribution on \mathcal{S} . At each time step, the agent takes an *action* $A_t \in \mathcal{A}$, receives feedback in terms of a real-valued *reward* $R_t \sim \mathcal{R}(S_t, A_t)$ and transitions to a new state $S_{t+1} \sim \mathcal{P}(S_t, A_t)$. The *random return* is the discounted cumulative sum of rewards received by the agent from the initial state onwards

$$G_t = \sum_{t=0}^{\infty} \gamma^t R_t.$$

A *stationary Markov policy* $\pi : \mathcal{S} \rightarrow \mathcal{P}(\mathcal{A})$ is a mapping from states to distributions over actions, describing a particular way of interacting with the environment such that

$$A_t \sim \pi(\cdot | S_t).$$

We denote the set of all policies by Π . The quality of a policy is measured by the its *expected return*

$$\mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t R_t \right].$$

For any policy $\pi \in \Pi$, the *value function* $V^{\pi}(s)$ measures the expected return received when starting from an initial state $s \in \mathcal{S}$ and acting according to π :

$$V^{\pi}(s) := \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t R_t \mid S_0 = s \right].$$

The upper-bound value is $V_{\max} := \frac{R_{\max}}{1-\gamma}$. Importantly, the value function of a state can be expressed with the immediate action A_0 , reward R_0 and the next state S_1 . This recursive relationship is called Bellman's equation [Bellman, 1957]

$$V^{\pi}(s) = \mathbb{E}_{\pi} [R_0 + \gamma V^{\pi}(S_1) \mid S_0 = s].$$

Similarly, the *action-value function* of a state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$ is defined as

$$Q^{\pi}(s, a) := \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t R_t \mid S_0 = s, A_0 = a \right].$$

Throughout this thesis, we will find it convenient to express these equations in vector notation [Puterman, 1994].

In the control setting, we are interested in finding an *optimal* policy π^* , that is which maximizes the expected return at every state

$$V^{\pi^*}(s) \geq V^\pi(s) \text{ for all } s \in \mathcal{S}.$$

Following Bellman’s principle of optimality, the optimal action-value function Q^* , corresponding to the optimal policy π^* , also satisfies Bellman’s equation

$$Q^*(s, a) = \mathbb{E}_\pi \left[R_0 + \gamma \max_{a' \in \mathcal{A}} Q^*(S_1, a') \mid S_0 = s, A_0 = a \right]$$

Learning Q^* is at the heart of *value-based methods*, algorithms on which we focus in this thesis.

2.2 Representation Learning

In large scale or continuous reinforcement learning problems, it is not possible to rely on a tabular representation of the value function and dynamic programming methods. Instead, it is common to use a structured representation of the state space from which we can express the value function in a more informative way. It allows to parametrize the value function with few parameters shared across states and also to generalize value predictions to new states. The quality of policies learnt through value function approximation depends on the choice of this state representation.

2.2.1 Feature Engineering in Classical Reinforcement Learning

Prior work on representation learning mainly focused on encoding a fairly exhaustive list of features and feeding these directly to the RL agent.

Linear value function approximation is a common function approximation approach [Tsitsiklis and Van Roy, 1996, Boyan, 2002, Sutton et al., 2008] that consists in representing the value function as a linear combination of *basis functions*, also referred to as features. It can be easily implemented, is interpretable and offers

many theoretical guarantees [see e.g. Sutton and Barto, 1998]. Given a mapping $\phi : \mathcal{S} \rightarrow \mathbb{R}^d$, a linear value function approximation $V_{\phi,w}$ is defined by

$$V_{\phi,w}(s) = \phi(s)^\top w,$$

where $w \in \mathbb{R}^d$ is a weight vector. In general, we are interested in the setting where the number of features is much less than the number of states $d \ll |\mathcal{S}|$.

In early applications, the weights $w \in \mathbb{R}^d$ were learnt from data by approximate dynamic programming [De Farias, 2003, Guestrin et al., 2003], temporal difference learning [Sutton and Barto, 1998, Tsitsiklis and Van Roy, 1996] or linear least square temporal difference [Bradtke and Barto, 1996] but the feature vectors $\phi(s)$ were usually hand-engineered [Samuel, 1959]. Designing these features is usually domain-specific and time-consuming, hence the resulting basis functions do not scale to large complex environments.

State abstraction is a simple way to construct binary state features by aggregating states according to some notion of similarity [Li et al., 2006]. The underlying idea is that a useful representation should ignore task-irrelevant information. Bisimulation [Givan et al., 2003, Ferns et al., 2004] is an example of a state abstraction where states are clustered together when they agree on immediate rewards and transition to groups of states also judged similar.

Examples of common *parametric families* of basis functions that have been used include radial basis functions [RBF; Lagoudakis and Parr, 2003] and polynomials or Fourier basis [Konidaris et al., 2011]. The representation capacity can also be improved by exhaustively generating features and then performing dimensionality reduction. Tile coding is an example of this approach [Sutton, 1996]. However, these methods encounter challenges in accurately approximating some value functions because of their non linearities. For instance, Dayan [1993] showed on simple grid worlds that states that are close under a euclidean metric may have very different values. In contrast, linear approximation architectures such as RBFs assume a Euclidean geometry of the state space and do not make a distinction between reachable and unreachable states.

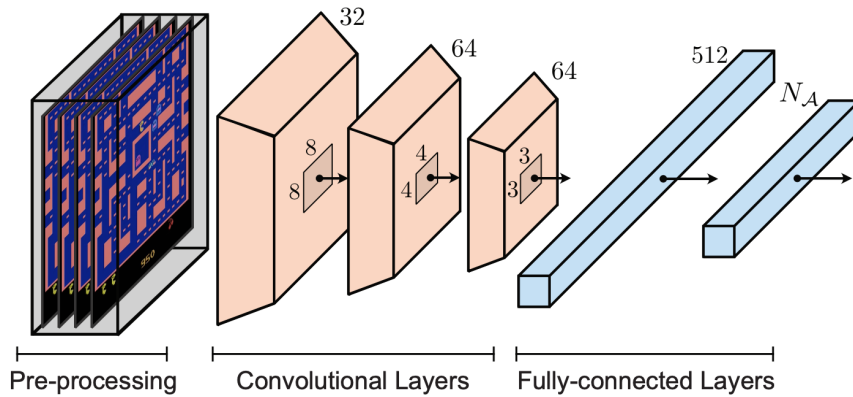


Figure 2.2: A deep RL architecture reproduced from Bellemare et al. [2023]

To overcome these limitations, a number of methods focused on learning basis functions automatically. Kretchmar and Anderson [1999] rely on a notion of temporal neighborhood to form new parametric basis functions. Menache et al. [2005], Kveton and Hauskrecht [2006] tune the parameters of parametric basis functions during the learning process by gradient descent or the Cross Entropy method. Keller et al. [2006], Petrik [2007], Parr et al. [2008] generate basis functions using the Bellman error of a current value function approximation. In these approaches, the basis functions depend on the reward function of the MDP.

Several approaches used the underlying dynamics of the MDP to build state representations. The successor representation [Dayan, 1993] is a time-based representation that encodes the temporal proximity of states given the agent’s policy. Proto-value functions [Mahadevan and Maggioni, 2007] are non-parametric basis functions built from an eigendecomposition of the graph Laplacian induced by the state transitions of the MDP. These representations reflect the geometry of the environment and are reward-agnostic making them for instance appealing for transfer learning across MDPs.

2.2.2 You Are What You Predict: Representation Learning in Deep Reinforcement Learning

Deep learning is today’s method of choice to learn a state representation. A particularity of deep learning applied to reinforcement learning is that we start

from raw inputs or perceptual inputs that we feed through a number of layers in the *deep neural network* to learn some structure. This eventually becomes prediction of the value function or policy telling which action the agent should take. For instance, the DQN algorithm [Deep Q-Networks, Mnih et al., 2015] applies the tools of *deep reinforcement learning* to learn an agent that outperforms humans at playing Atari 2600 video games. It leverages a deep neural networks to approximate the action-value function in combination with a semi-gradient Q-learning update rule. In Figure 2.2, four preprocessed images from some Atari games are successively transformed by three convolutional neural networks [LeCun et al., 1995] and a fully-connected layer that applies both a linear transformation and a non-linear activation function to the output of the last convolutional layer. This results in a 512-dimensional vector that we call representation ϕ . This vector is then linearly transformed by some weights $w \in \mathbb{R}^{512}$ into a value function for each action.

Under this view of deep RL [Yu and Bertsekas, 2009, Levine et al., 2017, Bellemare et al., 2019], we can then write the value function approximation at a state $s \in \mathcal{S}$ as

$$V_{\phi,w}(s) = \phi(s)^\top w.$$

In contrast with pre-deep learning feature engineering, the mapping $\phi : \mathcal{S} \rightarrow \mathbb{R}^d$, parameterized by all the stacks of layers after the image but before the actual prediction, is jointly learnt together with the weights $w \in \mathbb{R}^d$ of the last layer. It has also been shown that making additional predictions along the value function led to better performance empirically [Jaderberg et al., 2017, Bellemare et al., 2017, Dabney et al., 2021]. In practice, a gradient step is performed with respect to the network’s parameters towards minimising a combination of the DQN loss and the auxiliary loss. A hypothesis is that these *auxiliary tasks* lead to richer representations by predicting many aspects of the world. We now discuss some examples of auxiliary tasks that have been used in the literature.

The UNREAL algorithm makes auxiliary predictions about future pixel values [Jaderberg et al., 2017]. In addition, it also predicts future reward signals, similarly to Liu et al. [2021].

C51 [Bellemare et al., 2017], QR-DQN [Dabney et al., 2018b] and IQN [Dabney et al., 2018a] rely on distributional RL and predict the return distribution instead of the expected return.

Recent works predict one’s own latent state representation multiple steps in the future [François-Lavet et al., 2019, Gelada et al., 2019, Schwarzer et al., 2021]. They demonstrate significant empirical improvements and better sample-efficiency on Atari games.

Bellemare et al. [2019] propose adversarial value functions (AVFs), a class of auxiliary tasks which aim to minimize the approximation error of any value function. The method builds on insights from Dadashi et al. [2019] who highlighted that the value function space is a polytope.

Finally, a line of work focuses on state similarity and consists in training a network on auxiliary predictions such that the distance between two latent states corresponds to a behavioral metric, for instance the π -bisimulation metric [Castro, 2020, Zhang et al., 2020, Agarwal et al., 2021a] or the MiCo distance [Castro et al., 2021, Le Lan and Agarwal, 2023]. Some benefits of these approaches include the interesting theoretical properties they induce on the latent state space such as the Lipschitz continuity of the value function.

2.2.3 Representations Matter

With the observations from the previous section, it is now clear that the representation that RL agents use or learn matters a lot. This motivates the need to understand what a good representation for RL is.

As a first order approximation, consider an agent acting according to a policy π and the problem of estimating its value function by batch Monte Carlo. We are given a training dataset consisting of pairs of states and their value $D = \{(s_1, V^\pi(s_1)), \dots, (s_n, V^\pi(s_n))\} \in (\mathcal{S} \times \mathbb{R})^n$ and want to learn an approximation of the true value function V^π . Under the deep RL model described in Subsection 2.2.2, the aim is to solve the following optimization problem

$$\min_{\phi} \min_{w \in \mathbb{R}^d} \mathbb{E}_{s \sim \xi} \left[\left(\phi(s)^\top w - V^\pi(s) \right)^2 \right]. \quad (2.1)$$

Here, ϕ is parameterized by means of a neural network and jointly learnt together with the weights $w \in \mathbb{R}^d$.

A one-dimensional trivial "value-as-feature" representation performs very well on this problem, assuming the network is other unconstrained. Indeed, the choice $\phi(s) = V^\pi(s)$ for all states $s \in \mathcal{S}$ and $w = 1$ achieves zero error.

Now, at the other extreme, consider a tabular representation $\phi(s) = [\mathbb{I}_{[s=s']}]_{s' \in \mathcal{S}}$ where every state gets assigned a one hot encoding. From the perspective of minimising the error Equation (2.1) above, this representation is exactly equivalent to the "value-as-feature" representation, as there exists a weight vector such that it achieves zero error.

Yet, these two representations are very different. It is intuitive that they are not very satisfying and induce different behaviors in (deep) reinforcement learning. This leads us to look at state representations with quantities other than the approximation error from Equation (2.1). This example also suggests that, even if the value function is the same, the state feature plays a major role in adjacent things to learning the value function itself. The nature of a good representation in RL has been characterized from different perspectives in the literature.

Quality of approximation. Bellemare et al. [2019] consider the quality of approximation of the value function for all stationary policies given an MDP. They call a representation optimal when a solution to the following representation learning problem

$$\min_{\phi \in \mathcal{R}} \max_{\pi \in \mathcal{P}} \left\| \hat{V}_\phi^\pi - V^\pi \right\|_2^2.$$

Learning dynamics. Ghosh and Bellemare [2020] investigate representation learning under the lens of stability. They find that the Schur decomposition of the transition matrix guarantees the stability of TD(0) with linear value approximation. This representation can be learnt by a neural network using stochastic gradient descent on an auxiliary task update rule.

Transfer to other policies. Dabney et al. [2021] argue that a good representation allows for the good approximation error of the value function of a set of interesting policies. In particular, it transfers well to policies along the value improvement path.

Exploration. In the problem of exploration, agents should visit states that are reachable but have rarely been visited. Machado et al. [2018] demonstrate the usefulness of time-based state representation as a learning signal, to explore complex, sparse reward environments. In particular, they introduce an algorithm for option-based and count-based exploration relying on the successor representation. Burda et al. [2018] propose an exploration bonus which is the error predicting the learnt representation.

3

Metrics and Continuity in Reinforcement Learning

Abstract

In most practical applications of reinforcement learning, it is untenable to maintain direct estimates for individual states; in continuous-state systems, it is impossible. Instead, researchers often leverage *state similarity* (whether explicitly or implicitly) to build models that can generalize well from a limited set of samples. The notion of state similarity used, and the neighbourhoods and topologies they induce, is thus of crucial importance, as it will directly affect the performance of the algorithms. Indeed, a number of recent works introduce algorithms assuming the existence of “well-behaved” neighbourhoods, but leave the full specification of such topologies for future work. In this paper we introduce a unified formalism for defining these topologies through the lens of metrics. We establish a hierarchy amongst these metrics and demonstrate their theoretical implications on the Markov Decision Process specifying the reinforcement learning problem. We complement our theoretical results with empirical evaluations showcasing the differences between the metrics considered.

3.1 Introduction

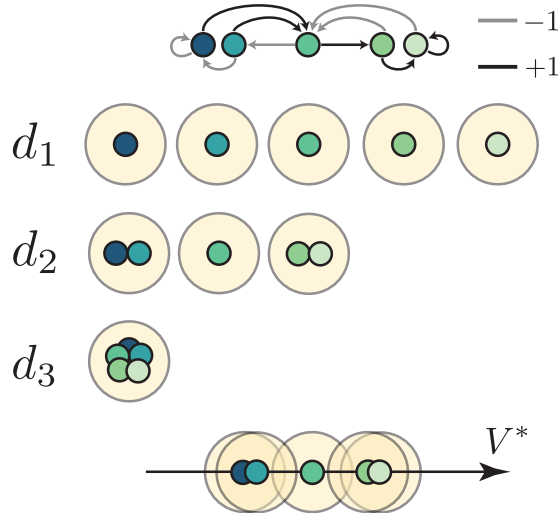


Figure 3.1: A simple five-state MDP (top) with the neighbourhoods induced by three metrics: an identity metric which isolates each state (d_1); a metric which captures behavioral proximity (d_2); and a metric which is not able to distinguish states (d_3). The yellow circles represent ϵ -balls in the corresponding metric spaces. The bottom row indicates the V^* values for each state.

A simple principle to generalization in reinforcement learning is to require that similar states be assigned similar predictions. State aggregation implements a coarse version of this principle, by using a notion of similarity to group states together. A finer implementation is to use the similarity in an adaptive fashion, for example by means of a nearest neighbour scheme over representative states. This approach is classically employed in the design of algorithms for continuous state spaces, where the fundamental assumption is the existence of a metric characterizing the real-valued distance between states.

To illustrate this idea, consider the three similarity metrics depicted in Figure 3.1. The metric d_1 isolates each state, the metric d_3 groups together all states, while the metric d_2 aggregates states based on the similarity in their *long-term dynamics*. In terms of generalization, d_1 would not be expected to generalize well as new states cannot leverage knowledge from previous states; d_3 can cheaply generalize to new states, but at the expense of accuracy; on the other hand, d_2 seems to strike a good balance between the two extremes.

In this paper we study the effectiveness of *behavioural metrics* at providing a good notion of state similarity. We call behavioural metrics the class of metrics derived from properties of the environment, typically measuring differences in reward and transition functions. Since the introduction of bisimulation metrics [Ferns et al., 2004, 2005], a number of behavioural metrics have emerged with additional desirable properties, including lax bisimulation [Taylor et al., 2009, Castro and Precup, 2010] and π -bisimulation metrics [Castro, 2020]. Behavioural metrics are of particular interest in the context of understanding generalization, since they directly encode the differences in action-conditional outcomes between states, and hence allow us to make meaningful statements about the relationship between these states.

We focus on the interplay between behavioural metrics and the continuity properties they induce on various functions of interest in reinforcement learning. Returning to our example, V^* is only continuous with respect to d_1 and d_2 . The continuity of a set of functions (with respect to a given metric) is assumed in most theoretical results for continuous state spaces, such as uniform continuity of the transition function [Kakade et al., 2003]; Lipschitz continuity of all Q-functions of policies [Pazis and Parr, 2013], Lipschitz continuity of the rewards and transitions [Zhao and Zhu, 2014, Ok et al., 2018] or of the optimal Q-function [Song and Sun, 2019, Touati et al., 2020, Sinclair et al., 2019]. We find that behavioural metrics support these algorithms to varying degrees: the original bisimulation metric, for example, provides fewer guarantees than what is required by some near-optimal exploration algorithms [Pazis and Parr, 2013]. These results are particularly significant given that behavioural metrics form a relatively privileged group: any metric that enables generalization must in some sense reflect the structure of interactions within the environment and hence, act like a behavioural metric.

3.2 Overview

Our aim is to unify representations of state spaces and the notion of continuity via a taxonomy of metrics.

Our first contribution is a general result about the continuity relationships of different functions of the MDP (Theorem 1). While Gelada et al. [2019] (resp. Norets [2010]) proved the uniform Lipschitz continuity of the optimal action-value function (resp. local continuity of the optimal value function) given the uniform Lipschitz continuity (resp. local continuity) of the reward and transition functions and Rachelson and Lagoudakis [2010] showed the uniform Lipschitz continuity of the value function given the uniform Lipschitz continuity of the action-value function in the case of deterministic policies, Theorem 1 is a more comprehensive result about the different components of the MDP (reward and transition functions, value and action value functions), for a spectrum of continuity notions (local and uniform continuity, local and uniform Lipschitz continuity) and applicable with stochastic policies, also providing counterexamples demonstrating that these relationships are only implication results.

Our second contribution is to demonstrate that different metrics lead to different notions of continuity for different classes of functions (Subsection 3.5.1, Subsection 3.5.2 and Table 3.2). We first study metrics that have been introduced in the literature (presented in Subsection 3.3.2). While Li et al. [2006] provide a unified treatment of some of these metrics, they do not analyse these abstractions through the lens of continuity. Using our taxonomy, we find that most commonly discussed metrics are actually poorly suited for algorithms that convert representations into values, so we introduce new metrics to overcome this shortcoming (Subsection 3.5.2). We also analyse the relationships between the topologies induced by all the metrics in our taxonomy (Theorem 2).

Finally, we present an empirical evaluation that supports our taxonomy and shows the importance of the choice of a neighbourhood in reinforcement learning algorithms (Section 3.6).

3.3 Background

We consider an agent interacting with an environment, modelled as a Markov Decision Process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$ [Puterman, 1994]. Here \mathcal{S} is a

continuous state space with Borel σ -algebra Σ and \mathcal{A} a discrete set of actions. Denoting $\Delta(X)$ to mean the probability distribution over X , we also have that $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ is the transition function, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow [0, R_{\max}]$ is the measurable reward function, and $\gamma \in [0, 1)$ is the discount factor. We write \mathcal{P}_s^a to denote the next-state distribution over \mathcal{S} resulting from selecting action a in s and write \mathcal{R}_s^a for the corresponding reward.

A stationary policy $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ is a mapping from states to distributions over actions, describing a particular way of interacting with the environment. We denote the set of all policies by Π . For any policy $\pi \in \Pi$, the value function $V^\pi(s)$ measures the expected discounted sum of rewards received when starting from state $s \in \mathcal{S}$ and acting according to π :

$$V^\pi(s) := \mathbb{E} \left[\sum_{t \geq 0} \gamma^t \mathcal{R}_{S_t}^{A_t}; S_0 = s, A_t \sim \pi(\cdot | S_t) \right].$$

The maximum attainable value is $V_{\max} := \frac{R_{\max}}{1-\gamma}$. The value function satisfies Bellman's equation:

$$V^\pi(s) = \mathbb{E}_{A \sim \pi(\cdot | s)} [\mathcal{R}_s^A + \gamma \mathbb{E}_{S' \sim \mathcal{P}_s^A} V^\pi(S')].$$

The state-action value function or Q-function Q^π describes the expected discounted sum of rewards when action $a \in \mathcal{A}$ is selected from the starting state s , and satisfies the recurrence

$$Q^\pi(s, a) = \mathcal{R}_s^a + \gamma \mathbb{E}_{S' \sim \mathcal{P}_s^a} V^\pi(S').$$

A policy π is said to be optimal if it maximizes the value function at all states:

$$V^\pi(s) = \max_{\pi' \in \Pi} V^{\pi'}(s) \text{ for all } s \in \mathcal{S}.$$

The existence of an optimal policy is guaranteed in both finite and infinite state spaces. We will denote this policy $\pi^* \in \Pi$. The corresponding value function and Q-function are denoted respectively V^* and Q^* .

3.3.1 Metrics, Topologies, and Continuity

We begin by recalling standard definitions regarding metrics and continuity, two concepts central to our work.

Definition 1 (Royden, 1968). A **metric space** $\langle X, d \rangle$ is a nonempty set X of elements (called points) together with a real-valued function d defined on $X \times X$ such that for all x, y , and z in X : $d(x, y) \geq 0$; $d(x, y) = 0$ if and only if $x = y$; $d(x, y) = d(y, x)$ and $d(x, y) \leq d(x, z) + d(z, y)$. The function d is called a **metric**. A **pseudo-metric** d is a metric with the second condition replaced by the weaker condition $x = y \implies d(x, y) = 0$.

In what follows, we will often use *metric* to stand for *pseudo-metric* for brevity.

A metric d is useful for our purpose as it quantifies, in a real-valued sense, the relationship between states of the environment. Given a state s , a natural question is: What other states are similar to it? The notion of a *topology* gives a formal answer.

Definition 2 (Sutherland, 2009). A metric space $\langle X, d \rangle$ induces a **topology** (X, \mathcal{T}_d) defined as the collection of open subsets of X ; specifically, the subsets $U \subset X$ that satisfy the property that for each $x \in U$, there exists $\epsilon > 0$ such that the ϵ -neighbourhood $B_d(x, \epsilon) = \{y \in X \mid d(y, x) < \epsilon\} \subset U$.

Let (X, \mathcal{T}) and (X, \mathcal{T}') be two topologies on the same space X . We say that \mathcal{T} is **coarser** than \mathcal{T}' , or equivalently that \mathcal{T}' is **finer** than \mathcal{T} , if $\mathcal{T} \subset \mathcal{T}'$.

Given two similar states under a metric d , we are interested in knowing how functions of these states behave. In the introductory example, we asked specifically: how does the optimal value function behave for similar states? This leads us to the notion of functional continuity. Given $f : X \rightarrow Y$ a function between a metric space (X, d_X) and a metric space (Y, d_Y) ,

- **Local continuity (LC)**: f is locally continuous at $x \in X$ if for any $\epsilon > 0$, there exists a $\delta_{x, \epsilon} > 0$ such that for all $x' \in X$, $d_X(x, x') < \delta_{x, \epsilon} \implies d_Y(f(x), f(x')) < \epsilon$. f is said to be locally continuous on X if it is continuous at every point $x \in X$.

- **Uniform continuity (UC):** f is uniformly continuous on X when given any $\epsilon > 0$, there exists $\delta_\epsilon > 0$ such that for all $x, x' \in X$, $d_X(x, x') < \delta_\epsilon \implies d_Y(f(x), f(x')) < \epsilon$.
- **Local Lipschitz continuity (LLC):** f is locally Lipschitz continuous at $x \in X$ if there exists $\delta_x > 0, K_x > 0$ such that for all $x', x'' \in B_{d_X}(x, \delta_x)$, $d_Y(f(x'), f(x'')) \leq K_x d_X(x', x'')$.
- **Uniform Lipschitz continuity (ULC):** f is uniformly Lipschitz continuous if there exist $K > 0$ such that for all $x, x' \in X$ we have $d_Y(f(x), f(x')) \leq K d_X(x, x')$.

The relationship between these different forms of continuity is summarized by the following diagram:

$$\begin{array}{ccc}
 UC & \longleftarrow & ULC \\
 \downarrow & & \downarrow \\
 LC & \longleftarrow & LLC
 \end{array} \tag{3.1}$$

where an arrow indicates implication; for example, any function that is ULC is also UC.

Here, we are interested in functions of states and state-action pairs. Knowing whether a particular function f possesses some continuity property p under a metric d informs us on how well we can extrapolate the value $f(s)$ to other states; in other words, it informs us on the generalization properties of d .

3.3.2 Prior Metrics and Abstractions

The simplest structure is to associate states to distinct groups, what is often called state aggregation [Bertsekas, 2011]. This gives rise to an *equivalence relation*, which we interpret as a *discrete pseudo-metric*, that is a metric taking a countable range of values.

Definition 3. *An equivalence relation $E \subseteq X \times X$ induces a **discrete pseudo-metric** e^E where $e^E(x, x') = 0$ if $(x, x') \in E$, and 1 otherwise.*

\mathbf{f}	$\phi_{f,\eta}$
Q^*	approximate Q function abstraction ($\eta \geq 0$) / Q^* -irrelevance ($\eta = 0$)
\mathcal{R} and \mathcal{P}	approximate model abstraction ($\eta \geq 0$) / Model-irrelevance ($\eta = 0$)
Q^π	Q^π -irrelevance abstraction ($\eta = 0$)
$\max_{\mathcal{A}} Q^*$	a^* -irrelevance abstraction ($\eta = 0$)

Table 3.1: Different types of state abstractions.

Throughout the text, we will use e to denote discrete pseudo-metrics. Two extremal examples of metrics are the **identity metric** $e^{\mathbb{I}} : \mathcal{S} \times \mathcal{S} \rightarrow \{0, 1\}$, induced by the *identity relation* $\mathbb{I} = \{(s, t) \in \mathcal{S} \times \mathcal{S} \mid s = t\}$ (e.g. d_1 in Figure 3.1), and the **trivial metric** $e^{\mathbb{T}} : \mathcal{S} \times \mathcal{S} \rightarrow \{0\}$ that collapses all states together (e.g. d_3 in Figure 3.1).

In-between these extremes, η -abstractions [Li et al., 2006, Abel et al., 2016] are functions $\phi : \mathcal{S} \rightarrow \hat{\mathcal{S}}$ that aggregates states which are mapped close to each other by a function f . That is, given a threshold $\eta \geq 0$ and $f : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, $\phi_{f,\eta}(s) = \phi_{f,\eta}(t) \implies |f(s, a) - f(t, a)| \leq \eta$. We list a few choices for f along with the name of the abstraction we will refer to throughout this text in Table 3.1.

η -abstractions are defined in terms of a particular function of direct relevance to the agent. However, it is not immediately clear whether these abstractions are descriptive, and, more specifically, the kind of continuity properties they support. An alternative is to relate states based on the outcomes that arise from different choices, starting in these states. These are *bisimulation relations* [Givan et al., 2003].

Definition 4. *An equivalence relation $E \subseteq \mathcal{S} \times \mathcal{S}$ with \mathcal{S}_E the quotient space and $\Sigma(E)$ the Σ measurable sets closed under E , if whenever $(s, t) \in E$ we have:*

- **Bisimulation relation**[Givan et al., 2003].

*Behavioral indistinguishability under equal actions; namely, for any action $a \in \mathcal{A}$, $\mathcal{R}_s^a = \mathcal{R}_t^a$, and $\mathcal{P}_s^a(X) = \mathcal{P}_t^a(X)$ for all $X \in \Sigma(E)$. We call E a **bisimulation relation**. We denote the largest bisimulation relation as \sim , and its corresponding discrete metric as e^\sim .*

- **Lax-bisimulation relation** [Taylor et al., 2009].

Behavioral indistinguishability under matching actions; namely, for any action $a \in \mathcal{A}$ from state s there is an action $b \in \mathcal{A}$ from state t such that $\mathcal{R}_s^a = \mathcal{R}_t^b$, and $\mathcal{P}_s^a(X) = \mathcal{P}_t^b(X)$ for all $X \in \Sigma(E)$, and vice-versa, we call E a **lax-bisimulation relation**. We denote the largest lax-bisimulation relation as \sim_{lax} , and its corresponding discrete metric as $e^{\sim_{lax}}$.

- **π -bisimulation relation** [Castro, 2020]. Behavioral indistinguishability under a fixed policy; namely, given a policy $\pi \in \Pi$, $\sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}_s^a = \sum_{a \in \mathcal{A}} \pi(a|t) \mathcal{R}_t^a$, and $\sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}_s^a(X) = \sum_{a \in \mathcal{A}} \pi(a|t) \mathcal{P}_t^a(X)$ for all $X \in \Sigma(E)$. We call E a **π -bisimulation relation**. We denote the largest bisimulation relation as \sim_π , and its corresponding discrete metric as e^{\sim_π} .

A *bisimulation metric* is the continuous generalization of a bisimulation relation. Formally, d is a bisimulation metric if its kernel is equivalent to the bisimulation relation. The canonical bisimulation metric [Ferns et al., 2005] is constructed from the Wasserstein distance between probability distributions.

Definition 5. Let (Y, d_Y) be a metric space with Borel σ -algebra Σ . The Wasserstein distance [Villani, 2008] between two probability measures P and Q on Y , under a given metric d_Y is given by $W_{d_Y}(P, Q) = \inf_{\lambda \in \Gamma(P, Q)} \mathbb{E}_{(x, y) \sim \lambda} [d_Y(x, y)]$, where $\Gamma(P, Q)$ is the set of couplings between P and Q .

Lemma 1 (Ferns et al., 2005). Let \mathcal{M} be the space of state pseudo-metrics and define the functional $F : \mathcal{M} \rightarrow \mathcal{M}$ as $F(d)(x, y) = \max_{a \in \mathcal{A}} (|\mathcal{R}_x^a - \mathcal{R}_y^a| + \gamma W_d(\mathcal{P}_x^a, \mathcal{P}_y^a))$. Then F has a least fixed point d^\sim and d^\sim is a bisimulation metric.

In words, bisimulation metrics arise as the fixed points of an operator on the space of pseudo-metrics. Lax bisimulation metrics $d^{\sim_{lax}}$ and a π -bisimulation metrics d^{\sim_π} can be defined in an analogous fashion; for succinctness, their formal definitions are included in Appendix 3.C.

3.4 Continuity Relationships

Our first result characterizes the continuity relationships between key functions of the MDP. The theorem considers different forms of continuity and relates how the continuity of one function implies another. While the particular case of uniform Lipschitz continuity of Q^* (resp. local continuity of V^*) from $\mathcal{P} + \mathcal{R}$ has been remarked on before by Gelada et al. [2019] (resp. Norets [2010]) as well as the case of uniform Lipschitz continuity of V^π given the uniform Lipschitz continuity of Q^π for stochastic policies π [Rachelson and Lagoudakis, 2010], to the best of our knowledge this is the first comprehensive treatment of the topic, in particular providing counterexamples.

Theorem 1. *If we decompose the Cartesian product $\mathcal{S} \times \mathcal{A}$ as: $d_{\mathcal{S} \times \mathcal{A}}(s, a, s', a') = d_{\mathcal{S}}(s, s') + d_{\mathcal{A}}(a, a')$ with $d_{\mathcal{A}}$ the identity metric, the LC, UC and LLC relationships between \mathcal{P} , \mathcal{R} , V^π , V^* , Q^π and Q^* functions are given by diagram 3.2. A directed arrow $f \rightarrow g$ indicates that function g is continuous whenever f is continuous. Labels on arrows indicate conditions that are necessary for that implication to hold. $\mathcal{P} + \mathcal{R}$ is meant to stand for both \mathcal{P} and \mathcal{R} continuity; π -cont indicates continuity of $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$. An absence of a directed arrow indicates that there exists a counter-example proving that the implication does not exist. In the ULC case, the previous relationships also hold with the following additional assumptions: $\gamma L_{\mathcal{P}} < 1$ for $\mathcal{P} + \mathcal{R} \rightarrow Q^*$ and $\gamma L_{\mathcal{P}}(1 + L_\pi) < 1$ for $\mathcal{P} + \mathcal{R} \xrightarrow{\pi\text{-cont}} Q^\pi$ where $L_{\mathcal{P}}$ and L_π are the Lipschitz constants of \mathcal{P} and π , respectively.*

$$\begin{array}{ccc}
 & & Q^\pi \xrightarrow{\pi\text{-cont}} V^\pi \\
 & \nearrow^{\pi\text{-cont}} & \\
 \mathcal{P} + \mathcal{R} & & \\
 & \searrow & \\
 & & Q^* \longrightarrow V^*
 \end{array} \tag{3.2}$$

Proof. All proofs and counterexamples are provided in Appendix 3.A. \square

The arrows are transitive and apply for all forms of continuity illustrated in diagram 3.1; for example, if we have ULC for Q^* , this implies we have LC for V^* . This diagram is useful when evaluating metrics as they clarify the strongest (or weakest) form of continuity one can demonstrate. When considering deterministic policies, we can notice that the π -continuity mentioned in Theorem 1 is very restrictive, as the following lemma shows.

Lemma 2. *If a deterministic policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ is continuous, \mathcal{S} is connected¹ and \mathcal{A} is discrete, then π is globally constant.*

3.5 Taxonomy of Metrics

We now study how different metrics support the continuity of functions relevant to reinforcement learning and the relationship between their induced topologies. While the taxonomy we present here is of independent interest, it also provides a clear theoretical foundation on which to build results regarding metric-respecting embeddings [Gelada et al., 2019, Zhang et al., 2020].

3.5.1 Continuity: Prior Metrics

We begin the exposition by considering the continuity induced by discrete metrics. These enable us to analyze the properties of some representations found in the literature. The extremes of our metric hierarchy are the identity metric $e^{\mathbb{I}}$ and trivial metric $e^{\mathbb{T}}$, which respectively support all and one continuous functions, and were represented by d_1 and d_3 in the introductory example.

Lemma 3 (Identity metric). *$e^{\mathbb{I}}$ induces the finest topology on \mathcal{S} , made of all possible subsets of \mathcal{S} . Let (Y, d_Y) be any metric space. Any function h (resp. Any bounded h) : $(\mathcal{S}, e^{\mathbb{I}}) \rightarrow (Y, d_Y)$ is LC and UC (resp. ULC).*

¹A connected space is topological space that cannot be represented as the union of two or more disjoint non-empty open subsets.

Lemma 4 (Trivial metric). $e^{\mathbb{I}}$ induces the coarsest topology on \mathcal{S} , consisting solely of $\{\emptyset, \mathcal{S}\}$. Let (Y, d_Y) be any metric space. Any function $h : (\mathcal{S}, e^{\mathbb{I}}) \rightarrow (Y, d_Y)$ is LC, UC and ULC iff h is constant.

We can also construct a discrete metric from any state aggregation $\phi : \mathcal{S} \rightarrow \hat{\mathcal{S}}$ as $e^\phi(s, t) = e^{\mathbb{I}}(\phi(s), \phi(t)) = 0$ if $\phi(s) = \phi(t)$, and 1 otherwise. However, as stated below, η -abstractions do not guarantee continuity except in the trivial case where $\eta = 0$.

Lemma 5. If $\eta = 0$, then any function f (resp. bounded function f): $(\mathcal{S}, d_{\mathcal{S}}) \rightarrow (Y, d_Y)$ is LC and UC (resp. ULC) with respect to the pseudometric $e^{\phi_{f,\eta}}$. However, given a function f and $\eta > 0$, there exists an η -abstraction $\phi_{f,\eta}$ such that f is not continuous with respect to $e^{\phi_{f,\eta}}$.

Unlike the discrete metrics defined by η -abstractions, both bisimulation metrics and the metric induced by the bisimulation relation support continuity of the optimal value function.

Lemma 6. Q^* (resp. Q^π) is ULC with Lipschitz constant 1 with respect to d^\sim (resp. $d^{\sim\pi}$).

Corollary 1. Q^* (resp. Q^π) is ULC with Lipschitz constant V_{max} with respect to e^\sim (resp. $e^{\sim\pi}$).

We note that Ferns et al. [2004] proved a weaker statement involving V^* (resp. Castro et al. [2009], V^π). To summarize, metrics that are too coarse may fail to provide the requisite continuity of reinforcement learning functions. Bisimulation metrics are particularly desirable as they achieve both a certain degree of coarseness, while preserving continuity. In practice, however, Ferns et al.’s bisimulation metric is difficult to compute and estimate, and tends to be conservative – as long as two states can be distinguished by action sequences, bisimulation will keep them apart.

3.5.2 Value-Based Metrics

As an alternative to bisimulation metrics, we consider simple metrics constructed from value functions and study their continuity offerings. These metrics are simple in that they are defined in terms of differences between values, or functions of values, at the states being compared. The last metric, $d_{\Delta_{\vee}}$, is particularly appealing as it can be approximated, as we describe below. Under this metric, all Q -functions are Lipschitz continuous, supporting some of the more demanding continuous-state exploration algorithms [Pazis and Parr, 2013].

Lemma 7. *For a given MDP, let Q^{π} be the Q -function of policy π , and Q^* the optimal Q -function. The following are continuous pseudo-metrics:*

1. $d_{\Delta^*}(s, s') = \max_{a \in \mathcal{A}} |Q^*(s, a) - Q^*(s', a)|$
2. $d_{\Delta_{\pi}}(s, s') = \max_{a \in \mathcal{A}} |Q^{\pi}(s, a) - Q^{\pi}(s', a)|$
3. $d_{\Delta_{\vee}}(s, s') = \max_{\pi \in \Pi, a \in \mathcal{A}} |Q^{\pi}(s, a) - Q^{\pi}(s', a)|$

Q^* (resp. Q^{π}) is ULC with Lipschitz constant 1 wrt to d_{Δ^*} (resp. $d_{\Delta_{\pi}}$). Q^{π} is ULC with Lipschitz constant 1 wrt to $d_{\Delta_{\vee}}$ for any $\pi \in \Pi$.

Remark. *When \mathcal{S} is finite, the number of policies to consider to compute $d_{\Delta_{\vee}}$ is finite: $d_{\Delta_{\vee}}(s, s') = \max_{\pi \in \Pi, a \in \mathcal{A}} |Q^{\pi}(s, a) - Q^{\pi}(s', a)| = \max_{\pi \in \Pi_{AVF}, a \in \mathcal{A}} |Q^{\pi}(s, a) - Q^{\pi}(s', a)|$, where Π_{AVF} is the finite set of extremal policies corresponding to Adversarial Value Functions (AVFs) [Bellemare et al., 2019].*

$d_{\Delta_{\vee}}$ provides strong continuity of the value-function for all policies contrary to any other metric that has been used in the literature. Since computing $d_{\Delta_{\vee}}$ is computationally expensive, we will approximate it by the pseudometric $d_{\widetilde{AVF}(n)} = \max_{\pi \in \Pi_{\widetilde{AVF}(n)}, a \in \mathcal{A}} |Q^{\pi}(s, a) - Q^{\pi}(s', a)|$, where $\Pi_{\widetilde{AVF}(n)}$ are n samples from the set of extremal policies Π_{AVF} .

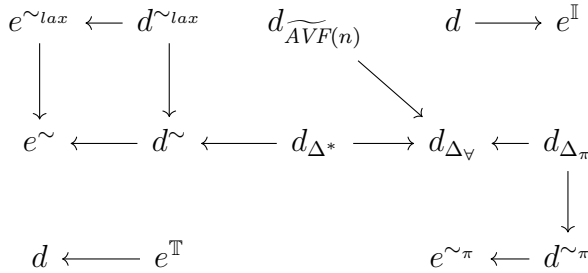
Metric	LC	UC	ULC	LLC	Complexity
Discrete metric $e^{\mathbb{I}}$	$Y^{\mathcal{S}}$	$Y^{\mathcal{S}}$	$\mathcal{B}(Y^{\mathcal{S}})$	$\mathcal{B}_L(Y^{\mathcal{S}})$	$O(\mathcal{S})$
Trivial metric $e^{\mathbb{T}}$	$\{y\}^{\mathcal{S}}$	$\{y\}^{\mathcal{S}}$	$\{y\}^{\mathcal{S}}$	$\{y\}^{\mathcal{S}}$	$O(1)$
Model-irrelevance	\mathcal{P}, \mathcal{R}	\mathcal{P}, \mathcal{R}	\mathcal{P}, \mathcal{R}	\mathcal{P}, \mathcal{R}	
Q^π -irrelevance	Q^π	Q^π	Q^π	Q^π	
Q^* -irrelevance	Q^*	Q^*	Q^*	Q^*	
a^* -irrelevance	Q^*	Q^*	Q^*	Q^*	
Approx. abstraction	-	-	-	-	
e^\sim	Q^*	Q^*	Q^*	Q^*	$O(\mathcal{A} \mathcal{S} ^3)$
d^\sim	Q^*	Q^*	Q^*	Q^*	$O(\mathcal{A} \mathcal{S} ^5 \log \mathcal{S} \frac{\ln \delta}{\ln \gamma})$
$e^{\sim\pi}$	Q^π	Q^π	Q^π	Q^π	$O(\mathcal{S} ^3)$
$d^{\sim\pi}$	Q^π	Q^π	Q^π	Q^π	$O(\mathcal{S} ^5 \log \mathcal{S} \frac{\ln \delta}{\ln \gamma})$
$e^{\sim lax}$	V^*	V^*	V^*	V^*	$O(\mathcal{A} ^2 \mathcal{S} ^3)$
$d^{\sim lax}$	V^*	V^*	V^*	V^*	$O(\mathcal{A} ^2 \mathcal{S} ^5 \log \mathcal{S} \frac{\ln \delta}{\ln \gamma})$
d_{Δ^*}	Q^*	Q^*	Q^*	Q^*	$O(\mathcal{S} ^2 \mathcal{A} ^{\frac{\log(\mathcal{R}_{\max}^{-1} \delta (1-\gamma))}{\log(\gamma)}})$
$d_{\Delta\pi}$	Q^π	Q^π	Q^π	Q^π	$O(\mathcal{S} ^2 \mathcal{A} ^{\frac{\log(\mathcal{R}_{\max}^{-1} \delta (1-\gamma))}{\log(\gamma)}})$
$d_{\Delta\forall}$	$Q^\pi, \forall \pi \in \Pi$	$Q^\pi, \forall \pi \in \Pi$	$Q^\pi, \forall \pi \in \Pi$	$Q^\pi, \forall \pi \in \Pi$	NP-hard?

Table 3.2: Categorization of state metrics, their continuity implications, and their complexity (when known). The notation $\{y\}^{\mathcal{S}}$ denotes any function $h : \mathcal{S} \rightarrow Y$ that is constant, $Y^{\mathcal{S}}$ refers to all functions $h : \mathcal{S} \rightarrow Y$. $\mathcal{B}(Y^{\mathcal{S}})$ (resp. $\mathcal{B}_L(Y^{\mathcal{S}})$) is a bounded (resp. locally bounded) function $h : \mathcal{S} \rightarrow Y$. “-” denotes an absence of LC, UC, ULC and LLC. In the complexity column, δ is the desired accuracy.

3.5.3 Categorizing Metrics, Continuity and Complexity

We now formally present in Theorem 2 the topological relationships between the different metrics. This hierarchy is important for generalization purposes as it provides a comparison between the shapes of different neighbourhoods which serve as a basis for RL algorithms on continuous state spaces.

Theorem 2. *The relationships between the topologies induced by the metrics in Table 3.2 are given by the following diagram. We denote by $d_1 \rightarrow d_2$ when $\mathcal{T}_{d_1} \subset \mathcal{T}_{d_2}$, that is, when \mathcal{T}_{d_1} is coarser than \mathcal{T}_{d_2} . Here d denotes any arbitrary metric.*



Proof. All proofs can be found in Appendix 3.B. The relation $d^{\sim lax} \rightarrow d^\sim$ was shown by Taylor et al. [2009] but not expressed in topological terms. \square

We summarize in Table 3.2 our continuity results mentioned throughout this section and supplement them with the continuity of the lax-bisimulation metric proven in Taylor et al. [2009]. To avoid over-cluttering the table, we only specify the strongest form of functional continuity according to Theorem 1. As an additional key differentiator, we also note the complexity of computing these metrics from a full model of the environment, which gives some indication about the difficulty of performing state abstraction. Proofs are provided in Appendix 3.B.

From a computational point of view, all continuous metrics can be approximated using deep learning techniques which makes them even more attractive to build representations. Atari 2600 experiments by Castro [2020] show that π -bisimulation metrics do perform well in larger domains. This is also supported by [Zhang et al., 2020] who use an encoder architecture to learn a representation that respects the bisimulation metric.

3.6 Empirical Evaluation

We now conduct an empirical evaluation to quantify the magnitude of the effects studied in the previous sections. Specifically, we are interested in how approximations derived from different metrics impact the performance of basic reinforcement learning procedures. We consider two kinds of approximations: state aggregation and nearest neighbour, which we combine with six representative metrics: e^\sim , $e^{\sim lax}$, d^\sim , $d^{\sim lax}$, d_{Δ^*} , and $d_{\widetilde{AVF}(50)}$.

We conduct our experiments on Garnet MDPs, which are a class of randomly generated MDPs [Archibald et al., 1995, Piot et al., 2014]. Specifically, a Garnet MDP $Garnet(n_S, n_A)$ is parameterized by two values: the number of states n_S and the number of actions n_A , and is generated as follows: **1.** The branching factor $b_{s,a}$ of each transition \mathcal{P}_s^a is sampled uniformly from $[1 : n_S]$. **2.** $b_{s,a}$ states are picked uniformly randomly from \mathcal{S} and assigned a random value in $[0, 1]$; these values are then normalized to produce a proper distribution \mathcal{P}_s^a . **3.** Each \mathcal{R}_s^a is sampled uniformly in $[0, 1]$. The use of Garnet MDPs grants us a less-biased comparison of

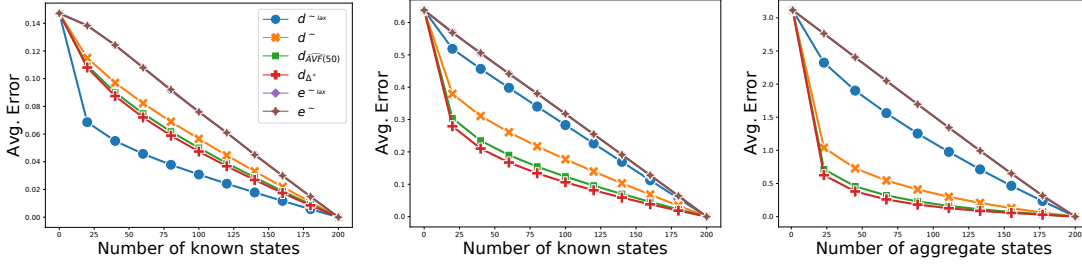


Figure 3.2: Errors when approximating the optimal value function (left) and optimal Q-function (center) via nearest-neighbours and errors when performing value iteration on aggregated states (right). Curves for e^{\sim} and $e^{\sim lax}$ are covering each other on all of the plots. Averaged over 100 Garnet MDPs with 200 states and 5 actions, with 50 independent runs for each (to account for subsampling differences). Confidence intervals were very tiny due to the large number of runs so were not included.

the different metrics than if we were to pick a few specific MDPs. Nonetheless, we do provide extra experiments on a set of GridWorld tasks in Appendix 3.D.

3.6.1 Generalizing the Value Function V^*

We begin by studying the approximation error that arises when extrapolating the optimal value function V^* from a subset of states. Specifically, given a subsampling fraction $f \in [0, 1]$, we sample $\lceil |\mathcal{S}| \times f \rceil$ states and call this set κ . For each unknown state $s \in \mathcal{S} \setminus \kappa$, we find its nearest known neighbour according to metric d : $NN(s) = \arg \min_{t \in \kappa} d(s, t)$. We then define the approximation error as $\hat{V}^*(s) = V^*(NN(s))$, and report the approximation error in Figure 3.2 (left). This experiment gives us insights into how amenable the different metrics are for transferring value estimates across states; effectively, their generalization capabilities.

According to Theorem 2, the two discrete metrics e^{\sim} and $e^{\sim lax}$ induce finer topologies than their four continuous counterparts. Most of the states being isolated from each other in these two representations, e^{\sim} and $e^{\sim lax}$ perform poorly. The three continuous metrics d^{\sim} , $d^{\sim lax}$ and d_{Δ^*} all guarantee Lipschitz continuity of V^* while $d_{\widetilde{AVF}(50)}$ is approximately V^* Lipschitz continuous. However, $d^{\sim lax}$ (resp. d_{Δ^*}) produce coarser (resp. approximately coarser) topologies than d^{\sim} (resp. $d_{\widetilde{AVF}(50)}$) (see Theorem 2). This is reflected in their better generalization error compared to the latter two metrics. Additionally, the lax bisimulation metric $d^{\sim lax}$ outperforms

d_{Δ^*} substantially, which can be explained by noting that $d^{\sim lax}$ measures distances between two states under independent action choices, contrary to all other metrics.

3.6.2 Generalizing the Q-function Q^*

We now illustrate the continuity (or absence thereof) of Q^* with respect to the different metrics. In Figure 3.2 (center), we perform a similar experiment as the previous one, still using a 1-nearest neighbour scheme but now extrapolating Q^* .

As expected, we find that metrics that do not support Q^* continuity, including $d^{\sim lax}$, cannot generalize from a subset of states, and their average error decreases linearly. In contrast, the three other metrics are able to generalize. Naturally, d_{Δ^*} , which aggregates states based on Q^* , performs particularly well. However, we note that $d_{\Delta_{\vee}}$ also outperforms the bisimulation metric d^{\sim} , highlighting the latter’s conservativeness, which tends to separate states more. By our earlier argument regarding $d^{\sim lax}$, this suggests there may be a class of functions, not represented in Table 3.2, which is continuous under d^{\sim} but not $d_{\widetilde{\text{AVF}}(50)}$.

3.6.3 Approximate Value Iteration

As a final experiment, we perform approximate value iteration using a state aggregation ϕ derived from one of the metrics. For each metric, we perform 10 different aggregations using a k -median algorithm, ranging from one aggregate state to 200 aggregate states. For a given aggregate state c , let $Q(c, a)$ stand for its associated Q-value. The approximation value iteration update is

$$\hat{Q}_k(c, a) \leftarrow \frac{1}{|c|} \sum_{s|\phi(s)=c} \left[\mathcal{R}_s^a + \gamma \mathbb{E}_{S' \sim \mathcal{P}_s^a} \max_{a \in \mathcal{A}} \hat{Q}_k(\phi(S')) \right]$$

We can then measure the error induced by our aggregation via

$$\max_{a \in \mathcal{A}} \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} |Q^*(s, a) - \hat{Q}_k(\phi(s), a)|,$$

which we display in the rightmost panel of Figure 3.2.

As in our second experiment, the metrics that do not support Q^* -continuity well fail to give good abstractions for approximate value iteration. As for e^{\sim} , the topology

induced by this metric is too fine (Theorem 2) leading to poor generalisation results. The performance of d_{Δ^*} is consistent with Theorem 2, which states that it induces the coarsest topology. However, although it is known that Q^* -continuity is sufficient for approximate value iteration [Li et al., 2006], it is somewhat surprising that it outperforms $d_{\widetilde{\text{AVF}}(50)}$, since $d_{\widetilde{\text{AVF}}(50)}$ is an approximant of $d_{\Delta_{\forall}}$ that is designed to provide continuity with respect to all policies, so it may be expected to yield better approximations at intermediate iterations. Despite this, $d_{\Delta_{\forall}}$ still serves as an interesting and tractable surrogate metric to d_{Δ^*} .

3.7 Discussion

Behavioral metrics are important both to evaluate the goodness of a given state representation and to learn such a representation. We saw that approximate abstractions and equivalence relations are insufficient for continuous-state RL problems, because they do not support the continuity of common RL functions or induce very fine representations on the state space leading to poor generalization.

Continuous behavioural metrics go one step further by considering the structure of the MDP in their construction and inducing coarser topologies than their discrete counterparts; however, within that class we still find that not all metrics are equally useful. The original bisimulation metric of Ferns et al. [2004], for example, is too conservative and has a rather fine topology. This is confirmed by our experiments in Figure 3.2, where it performs poorly overall. The lax bisimulation metric guarantees the continuity of V^* which makes it suitable for transferring optimal values between states but fails to preserve continuity of Q^* . Together with our analysis, the d_{Δ^*} and $d_{\Delta_{\forall}}$ metrics seem interesting candidates when generalising within a neighbourhood.

$d_{\Delta_{\forall}}$ is useful when we do not know the value improvement path the algorithm will be following [Dabney et al., 2021]. Despite being approximated from a finite number of policies, the performance of $d_{\widetilde{\text{AVF}}(n)}$, reflects the fact that it respects, in some sense, the entire space of policies that are spanned by policy iteration and makes it useful in practice. One advantage of this metric is that it is built from value functions, which are defined on a per-state basis; this makes it amenable to

online approximations. In contrast, bisimulation metrics are only defined for pairs of states, which makes it difficult to approximate in an online fashion, specifically due to the difficulty of estimating the Wasserstein metric on every update.

Finally, continuing our analysis on partially observable systems is an interesting area for future work. Although Castro et al. [2009] proposed various equivalence relations for partially observable systems, there has been little work in defining proper metrics for these systems.

Acknowledgements

The authors would like to thank Sheheryar Zaidi, Adam Foster and Abe Ng for insightful discussions about topology and functional analysis, Carles Gelada, John D. Martin, Dibya Ghosh, Ahmed Touati, Rishabh Agarwal, Marlos Machado and the whole Google Brain team in Montreal for helpful discussions, and Robert Dadashi for a conversation about polytopes. We also thank Lihong Li and the anonymous reviewers for useful feedback on this paper.

We would also like to thank the Python community [Van Rossum and Drake Jr, 1995, Oliphant, 2007] and in particular NumPy [Oliphant, 2006, Walt et al., 2011, Harris et al., 2020], Tensorflow [Abadi et al., 2016], SciPy [Jones et al., 2001], Matplotlib [Hunter, 2007] and Gin-Config (<https://github.com/google/gin-config>).

3.8 Broader Impact

This work lies in the realm of “foundational RL” in that it contributes to the fundamental understanding and development of reinforcement learning algorithms and theory. As such, despite us agreeing in the importance of this discussion, our work is quite far removed from ethical issues and potential societal consequences.

Function(s)	Domain	Range
\mathcal{P}	$\mathcal{S} \times \mathcal{A}$	$\Sigma \rightarrow [0, 1]$
\mathcal{R}	$\mathcal{S} \times \mathcal{A}$	$[0, R_{\max}] \subset \mathbb{R}$
V^π, V^*	\mathcal{S}	$[0, V_{\max}] \subset \mathbb{R}$
Q^π, Q^*	$\mathcal{S} \times \mathcal{A}$	$[0, V_{\max}] \subset \mathbb{R}$
π	\mathcal{S}	$\Delta(\mathcal{A})$

Table 3.3: RL functions with their respective domains and ranges.

3.A Proofs for Section 3.4

We begin by proving the first main theorem in the paper, Theorem 1. We report in Table 3.3 the domains and ranges of the different RL functions mentioned in Theorem 1 that will be used throughout the proof. Before proving this result, we introduce the following necessary lemma.

Lemma A. *Choosing the discrete topology on the finite space \mathcal{A} and assuming the product metric $d_{\mathcal{S} \times \mathcal{A}} = d_{\mathcal{S}} + d_{\mathcal{A}}$, the function $Q^\pi : (\mathcal{S} \times \mathcal{A}, d_{\mathcal{S} \times \mathcal{A}}) \rightarrow \mathbb{R}$ is continuous if and only if*

1. *The function $Q^\pi : (\mathcal{S} \times \{a\}, d_{\mathcal{S} \times \mathcal{A}}) \rightarrow \mathbb{R}$ is continuous for all $a \in \mathcal{A}$.*
2. *The function $Q^\pi(\cdot, a) : (\mathcal{S}, d_{\mathcal{S}}) \rightarrow \mathbb{R}$ is continuous for all $a \in \mathcal{A}$.*

Proof. To understand better the notion of continuity on the space \mathcal{A} endowed with the discrete topology, we refer the reader to Lemma 3.

We begin with the first equivalence.

(\implies) : For LC, UC, LLC and ULC, this result follows from the fact that the function on $\mathcal{S} \times \{a\}$ is a restriction of the function on $\mathcal{S} \times \mathcal{A}$. For instance in the case of LC, suppose Q^π is LC on $\mathcal{S} \times \mathcal{A}$. Let $\epsilon > 0$. Then, for all $(s, a), (s', a') \in \mathcal{S} \times \mathcal{A}$ there exists $\delta > 0$ such that, $d_{\mathcal{S} \times \mathcal{A}}((s, a), (s', a')) \leq \delta \implies |Q^\pi(s, a) - Q^\pi(s', a')| \leq \epsilon$. In particular, this is true for $a = a'$ so Q^π is LC on $\mathcal{S} \times \{a\}$ for all $a \in \mathcal{A}$.

(\impliedby) : Q^π is LC on $\mathcal{S} \times \{a\}$ for all $a \in \mathcal{A}$. So for all $a \in \mathcal{A}$, the limit of $Q(s_n, a)$ as the sequence $s_n \in \mathcal{S}$ converges to $s \in \mathcal{S}$ exists and is equal to $Q(s, a)$, that is $s_n \rightarrow s \implies Q(s_n, a) \rightarrow Q(s, a)$. Moreover, $(s_n, a_n) \rightarrow (s, a_0)$ implies that $a_n = a_0$ for all n big enough because \mathcal{A} has the discrete metric. So for

$n > N \in \mathbb{N}, Q(s_n, a_n) = Q(s_n, a_0) \rightarrow Q(s, a_0)$. Hence, $Q^\pi : (\mathcal{S} \times \{a\}, d_{\mathcal{S} \times \mathcal{A}}) \rightarrow \mathbb{R}$ is LC for all $a \in \mathcal{A} \implies Q^\pi : (\mathcal{S} \times \mathcal{A}, d_{\mathcal{S} \times \mathcal{A}}) \rightarrow \mathbb{R}$ is LC.

Now, in the UC case, Q^π is UC on $\mathcal{S} \times \{a\}$ for all $a \in \mathcal{A}$, so we have: $\forall a \in \mathcal{A}, \forall \epsilon > 0, \exists \delta_{\epsilon, a} > 0$, such that for all $s, s' \in \mathcal{S}, d_{\mathcal{S} \times \mathcal{A}}((s, a), (s', a)) \leq \delta_{\epsilon, a} \implies |Q^\pi(s, a) - Q^\pi(s', a)| < \epsilon$. The space \mathcal{A} being finite, $\min_{a \in \mathcal{A}} \delta_{\epsilon, a}$ exists and is positive. Hence, $\forall \epsilon > 0$, there exists $\delta_1 = \min \{ \min_{a \in \mathcal{A}} \delta_{\epsilon, a}, 1/2 \} > 0$, such that for all $(s, a), (s', a') \in \mathcal{S} \times \mathcal{A}$, if $d_{\mathcal{S} \times \mathcal{A}}((s, a), (s', a')) \leq \delta_1 < 1/2$, then $a = a'$ since $d_{\mathcal{A}}(a, a')$ can only take values 0 or 1. Applying UC of $Q^\pi(\cdot, a)$, we get $d_{\mathcal{S} \times \mathcal{A}}((s, a), (s', a')) \leq \delta_1 \leq \delta_{\epsilon, a} \implies |Q^\pi(s, a) - Q^\pi(s', a')| \leq \epsilon$. We can conclude that Q^π is UC on $\mathcal{S} \times \mathcal{A}$.

In the ULC case, Q^π is ULC on $\mathcal{S} \times \{a\}$ for all $a \in \mathcal{A}$, so we have: $\forall a \in \mathcal{A}, \exists L_a > 0$, such that for all $s, s' \in \mathcal{S}, |Q^\pi(s, a) - Q^\pi(s', a)| \leq L_a d_{\mathcal{S} \times \mathcal{A}}((s, a), (s', a))$. So, as Q^π is bounded by V_{\max} , there exists $L = \max \{ \max_{a \in \mathcal{A}} L_a, V_{\max} \} \geq 0$, such that for all $(s, a), (s', a') \in \mathcal{S} \times \mathcal{A}, |Q^\pi(s, a) - Q^\pi(s', a')| \leq L d_{\mathcal{S} \times \mathcal{A}}((s, a), (s', a')) = L(d_{\mathcal{S}}(s, s') + d_{\mathcal{A}}(a, a'))$.

In the LLC case, Q^π is LLC on $\mathcal{S} \times \{a\}$ for all $a \in \mathcal{A}$. So for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, there exists a neighbourhood U of $\mathcal{S} \times \mathcal{A}$ induced by $d_{\mathcal{S} \times \mathcal{A}}$ such that Q^π restricted to U is ULC. We conclude by the same argument as in the ULC case above.

The second equivalence is true because, for any $a \in \mathcal{A}$, the relabelling map $s \mapsto (s, a)$ is an isometry of metric spaces $(\mathcal{S} \times \{a\}, d_{\mathcal{S} \times \mathcal{A}})$ and $(\mathcal{S}, d_{\mathcal{S}})$. Isometric metric spaces are "equivalent" and hence have the same properties. \square

Theorem 1. *If we decompose the Cartesian product $\mathcal{S} \times \mathcal{A}$ as: $d_{\mathcal{S} \times \mathcal{A}}(s, a, s', a') = d_{\mathcal{S}}(s, s') + d_{\mathcal{A}}(a, a')$ with $d_{\mathcal{A}}$ the identity metric, the LC, UC and LLC relationships between $\mathcal{P}, \mathcal{R}, V^\pi, V^*, Q^\pi$ and Q^* functions are given by diagram 3.2. A directed arrow $f \rightarrow g$ indicates that function g is continuous whenever f is continuous. Labels on arrows indicate conditions that are necessary for that implication to hold. $\mathcal{P} + \mathcal{R}$ is meant to stand for both \mathcal{P} and \mathcal{R} continuity; π -cont indicates continuity of $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$. An absence of a directed arrow indicates that there exists a counter-example proving that the implication does not exist. In the ULC case, the previous relationships also hold with the following additional assumptions: $\gamma L_{\mathcal{P}} < 1$*

for $\mathcal{P} + \mathcal{R} \rightarrow Q^*$ and $\gamma L_{\mathcal{P}}(1 + L_{\pi}) < 1$ for $\mathcal{P} + \mathcal{R} \xrightarrow{\pi\text{-cont}} Q^{\pi}$ where $L_{\mathcal{P}}$ and L_{π} are the Lipschitz constants of \mathcal{P} and π , respectively.

$$\begin{array}{ccc}
 & & Q^{\pi} \xrightarrow{\pi\text{-cont}} V^{\pi} \\
 & \nearrow^{\pi\text{-cont}} & \\
 \mathcal{P} + \mathcal{R} & & \\
 & \searrow & \\
 & & Q^* \longrightarrow V^*
 \end{array} \tag{3.2}$$

The proof itself will be made up of a series of lemmas for each of the arrows (or lack thereof) in the diagram.

Lemma B. *If $Q^* : (\mathcal{S} \times \mathcal{A}, d_{\mathcal{S} \times \mathcal{A}}) \rightarrow \mathbb{R}$ is continuous, then $V^* : \mathcal{S} \rightarrow \mathbb{R}$ is continuous.*

By definition of the optimal value function,

$$V^*(s) = \max_{a \in \mathcal{A}} Q^*(s, a).$$

\mathcal{A} being a finite set of discrete actions and the *max* function being a non-expanding (that is, 1-Lipschitz) map, it results that if Q^* is LC (resp. UC, resp. ULC, resp. LLC) then V^* is LC (resp. UC, resp. ULC, resp. LLC).

In more details, let's assume Q^* is LC. Let $a \in \mathcal{A}$ and $\epsilon > 0$. By definition of LC of Q^* , there exists $\delta_{s,\epsilon}$ such that for all $s' \in \mathcal{S}$, $d_{\mathcal{S}}(s, s') \leq \delta_{s,\epsilon} \implies |Q^*(s, a) - Q^*(s', a)| \leq \epsilon$.

$$\begin{aligned}
 |V^*(s_1) - V^*(s_2)| &= \left| \max_{a \in \mathcal{A}} Q^*(s_1, a) - \max_{a \in \mathcal{A}} Q^*(s_2, a) \right| \\
 &\leq \max_{a \in \mathcal{A}} |Q^*(s_1, a) - Q^*(s_2, a)| \text{ as max is a non expansion.} \\
 &\leq \max_{a \in \mathcal{A}} \epsilon \text{ as } Q^* \text{ is LC.} \\
 &\leq \epsilon
 \end{aligned}$$

Hence, V^* is LC. The proof for the UC case is similar.

Now, in the ULC case: Let $s_1, s_2 \in \mathcal{S}$,

$$\begin{aligned} |V^*(s_1) - V^*(s_2)| &= \left| \max_{a \in \mathcal{A}} Q^*(s_1, a) - \max_{a \in \mathcal{A}} Q^*(s_2, a) \right| \\ &\leq \max_{a \in \mathcal{A}} |Q^*(s_1, a) - Q^*(s_2, a)| \text{ as max is a non expansion.} \\ &\leq \max_{a \in \mathcal{A}} L_a d(s_1, s_2) \text{ as } Q^* \text{ is ULC.} \\ &\leq Ld(s_1, s_2). \end{aligned}$$

We can thus conclude that V^* is also ULC. The LLC case is similar to the ULC proof.

The reverse implication is not true as shows the following counter-example. Suppose $\mathcal{S} = \mathbb{R}$, $\mathcal{A} = \{1, 2\}$ and $Q(s, 1) = 1$ for all s . And suppose $Q(s, 2)$ is some discontinuous function that is always less than 1. Let's for instance choose:

$$Q^*(s, 2) = \begin{cases} 0 & \text{if } s \leq s_0, s_0 \in \mathbb{R} \\ 0.5 & \text{if } s > s_0. \end{cases} \quad (3.3)$$

Then $V^*(s) = Q^*(s, 1)$ which is continuous but $Q^*(s, a)$ is not continuous at s_0 for $a = 2$.

Lemma C. *If $Q^\pi : (\mathcal{S} \times \mathcal{A}, d_{\mathcal{S} \times \mathcal{A}}) \rightarrow \mathbb{R}$ and $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ are continuous, then V^π is continuous.*

The value function V^π is defined as follows:

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) Q^\pi(s, a).$$

In the LC case: let's assume Q^π is LC. Let $a \in \mathcal{A}$ and $\epsilon > 0$. By definition of LC of Q^π , there exists $\delta_{s, \epsilon}$ such that for all $s' \in \mathcal{S}$, $d_{\mathcal{S}}(s, s') \leq \delta_{s, \epsilon} \implies |Q^\pi(s, a) - Q^\pi(s', a)| \leq \epsilon$. We also assume the policy is LC, that is, there exists $\delta'_{s, \epsilon}$ such that for all $s' \in \mathcal{S}$, $d_{\mathcal{S}}(s, s') \leq \delta'_{s, \epsilon} \implies W_{d_{\mathcal{A}}}(\pi(\cdot|s) - \pi(\cdot|s')) \leq \epsilon$. For

all $s' \in \mathcal{S}$ such that $d_{\mathcal{S}}(s, s') \leq \min(\delta'_{s,\epsilon}, \delta_{s,\epsilon})$, we have

$$\begin{aligned}
|V^\pi(s) - V^\pi(s')| &= |\mathbb{E}_{a \sim \pi(\cdot|s)} Q^\pi(s, a) - \mathbb{E}_{a \sim \pi(\cdot|s')} Q^\pi(s', a)| \\
&= |\mathbb{E}_{a \sim \pi(\cdot|s)} Q^\pi(s, a) - \mathbb{E}_{a \sim \pi(\cdot|s')} Q^\pi(s, a) \\
&\quad + \mathbb{E}_{a \sim \pi(\cdot|s')} Q^\pi(s, a) - \mathbb{E}_{a \sim \pi(\cdot|s')} Q^\pi(s', a)| \\
&\leq |\mathbb{E}_{a \sim \pi(\cdot|s)} Q^\pi(s, a) - \mathbb{E}_{a \sim \pi(\cdot|s')} Q^\pi(s, a)| \\
&\quad + |\mathbb{E}_{a \sim \pi(\cdot|s')} Q^\pi(s, a) - \mathbb{E}_{a \sim \pi(\cdot|s')} Q^\pi(s', a)| \\
&\leq |V_{\max} W_{d_{\mathcal{A}}}(\pi(\cdot|s) - \pi(\cdot|s'))| \\
&\quad + \mathbb{E}_{a \sim \pi(\cdot|s')} |Q^\pi(s, a) - Q^\pi(s', a)| \text{ by definition of the Wasserstein} \\
&\leq (V_{\max} + 1)\epsilon
\end{aligned}$$

So V^π is LC. The proof is similar in the UC case.

In the ULC case: let $s, s' \in \mathcal{S}$,

$$\begin{aligned}
|V^\pi(s) - V^\pi(s')| &= |\mathbb{E}_{a \sim \pi(\cdot|s)} Q^\pi(s, a) - \mathbb{E}_{a \sim \pi(\cdot|s')} Q^\pi(s', a)| \\
&= |\mathbb{E}_{a \sim \pi(\cdot|s)} Q^\pi(s, a) - \mathbb{E}_{a \sim \pi(\cdot|s')} Q^\pi(s, a) \\
&\quad + \mathbb{E}_{a \sim \pi(\cdot|s')} Q^\pi(s, a) - \mathbb{E}_{a \sim \pi(\cdot|s')} Q^\pi(s', a)| \\
&\leq |\mathbb{E}_{a \sim \pi(\cdot|s)} Q^\pi(s, a) - \mathbb{E}_{a \sim \pi(\cdot|s')} Q^\pi(s, a)| \\
&\quad + |\mathbb{E}_{a \sim \pi(\cdot|s')} Q^\pi(s, a) - \mathbb{E}_{a \sim \pi(\cdot|s')} Q^\pi(s', a)| \\
&\leq |V_{\max} W_{d_{\mathcal{A}}}(\pi(\cdot|s) - \pi(\cdot|s'))| \\
&\quad + \mathbb{E}_{a \sim \pi(\cdot|s')} |Q^\pi(s, a) - Q^\pi(s', a)| \text{ by definition of the Wasserstein} \\
&\leq V_{\max} L_\pi d(s, s') + \mathbb{E}_{a \sim \pi(\cdot|s')} \max_{a \in \mathcal{A}} L_a d(s, s') \\
&\text{as the policy and Q-functions are ULC} \\
&\leq (V_{\max} L_\pi + \max_{a \in \mathcal{A}} L_a) d(s, s')
\end{aligned}$$

We can thus conclude that V^π is also ULC with Lipschitz constant $(V_{\max} L_\pi + \max_{a \in \mathcal{A}} L_a)$. The same reasoning applies in the LLC case.

We emphasize that the continuity assumption of π is important, as the following example shows. Let's assume $\mathcal{S} = \mathbb{R}$ and $\mathcal{A} = \{1, 2\}$. Imagine we have the

following discontinuous policy:

$$\pi(a|s) = \begin{cases} \delta_0 & \text{if } s < 0 \\ \delta_1 & \text{if } s \geq 0. \end{cases} \quad (3.4)$$

where:

$$\delta_{x_0}(A) = \begin{cases} 1 & \text{if } x_0 \in A \\ 0 & \text{else.} \end{cases} \quad (3.5)$$

and the following value function:

$$Q^\pi(s, a) = \begin{cases} s & \text{if } a = 0 \\ s + 1 & \text{if } a = 1. \end{cases} \quad (3.6)$$

Then, V^π is discontinuous at 0.

$$V^\pi(s) = \begin{cases} s & \text{if } s < 0 \\ s + 1 & \text{if } s \geq 0. \end{cases} \quad (3.7)$$

It is clear that continuity of V^π does not imply continuity of Q^π : take the deterministic constant policy $\pi(a|s) = 1$ so that $V^\pi(s) = Q^\pi(s, 1)$. As previously, we can have any discontinuous function for $Q^\pi(s, 1)$.

Lemma 2. *If a deterministic policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ is continuous, \mathcal{S} is connected² and \mathcal{A} is discrete, then π is globally constant.*

Proof. π is continuous at s iff for all $\epsilon > 0$, there exists $\delta_{\epsilon, s} > 0$ such that for all $s' \in \mathcal{S}$, $d(s, s') \leq \delta_{\epsilon, s}$ implies $d_{\mathcal{A}}(\pi(s) - \pi(s')) \leq \epsilon$. In particular, choosing $\epsilon = \frac{1}{2}$ implies that $\pi(s) = \pi(s')$. π is thus locally constant. Supposing \mathcal{S} is connected implies π is globally constant. \square

While our proof above is valid for stochastic policies, we note that the proof of Lemma C in the ULC case with deterministic policies is provided by Rachelson and Lagoudakis [2010]:

Corollary 2 (Rachelson and Lagoudakis, 2010). *If Q^π is ULC with Lipschitz constant L_Q and the policy π is ULC with Lipschitz constant L_π , then V^π is ULC with Lipschitz constant $L_Q(1 + L_\pi)$.*

²A connected space is topological space that cannot be represented as the union of two or more disjoint non-empty open subsets.

Lemma D. *We assume that the next state probability measure \mathcal{P}_s^a admits a density $p_s^a : \mathcal{S} \rightarrow [0, \infty)$ with respect to the Lebesgue measure. If $s' \mapsto p_s^a(s')$ is bounded and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ and $s \mapsto p_s^a(s')$ are LC, then $Q^* : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is LC.*

We start by recalling the dominated convergence theorem:

Theorem 3. *(Lebesgue's Dominated Convergence Theorem)*

Let $\{f_n\}$ be a sequence of complex-valued measurable functions on a measure space $(\mathcal{S}, \Sigma, \mu)$. Suppose that:

1. the sequence $\{f_n\}$ converges pointwise to a function f
2. the sequence $\{f_n\}$ is dominated by some integrable function g , that is, $\forall n \in \mathbb{N}, \forall x \in \mathcal{S}, |f_n(x)| \leq g(x)$

Then, f is integrable and $\lim_{n \rightarrow \infty} \int_{\mathcal{S}} f_n(d\mu) = \int_{\mathcal{S}} f d\mu$.

Let's define the following sequence: $Q_0(s, a) = 0$ and $Q_{n+1}(s, a) = \mathcal{R}(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)}[\max_{a' \in \mathcal{A}} Q_n(s', a')]$ for all $s \in \mathcal{S}, a \in \mathcal{A}$.

Q_0 is constant and thus continuous on $\mathcal{S} \times \mathcal{A}$.

To show that the continuity of Q_n implies the continuity of Q_{n+1} , let's apply the dominated convergence theorem.

1. $s \mapsto p_s^a(s')$ being continuous, s_m tends to $s \in \mathcal{S}$ implies that

$$\max_{a' \in \mathcal{A}} [Q_n(s', a')] p_{s_m}^a(s') \text{ tends to } \max_{a' \in \mathcal{A}} [Q_n(s', a')] p_s^a(s').$$

2. For all $s' \in \mathcal{S}, |\max_{a'} Q_n(s', a')| < V_{max}$ by assumption. We fix $a \in \mathcal{A}$. p_s^a is bounded so there exists a function $h_a \in L^1(\mathcal{S})$ such that $p_s^a(s') \leq h_a(s')$ for all s', s .

By the dominated convergence theorem,

$$\lim_{m \rightarrow \infty} \int_{\mathcal{S}} \max_{a' \in \mathcal{A}} [Q_n(s', a')] p_{s_m}^a(s') ds' = \int_{\mathcal{S}} \max_{a' \in \mathcal{A}} [Q_n(s', a')] p_s^a(s') ds'.$$

If $s_m \rightarrow s$, then $\mathbb{E}_{s' \sim \mathcal{P}(\cdot|s_m,a)}[\max_{a' \in \mathcal{A}} Q_n(s', a')] \rightarrow \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)}[\max_{a' \in \mathcal{A}} Q_n(s', a')]$.

The reward function being LC by assumption, Q_{n+1} is LC.

Let's now show that $Q^* = \lim_{n \rightarrow \infty} Q_n$ is LC. Let $T : (C(\mathcal{S} \times \mathcal{A}), \|\cdot\|_\infty) \rightarrow (C(\mathcal{S} \times \mathcal{A}), \|\cdot\|_\infty)$, where $C(\mathcal{S} \times \mathcal{A}) = \mathcal{X}$ is the space of LC functions on $\mathcal{S} \times \mathcal{A}$ and $\|\cdot\|_\infty = \sup_{(s,a) \in \mathcal{S} \times \mathcal{A}}$, be defined by:

$$(Tf)(s, a) = \mathcal{R}(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)} [\max_{a' \in \mathcal{A}} f(s', a')].$$

It is known that $\|Tf - Tg\|_\infty \leq \gamma \|f - g\|_\infty$. The contraction mapping theorem implies that $TQ_n = Q_{n+1} \rightarrow Q^*$ in \mathcal{X} (sup norm), so $Q^* \in \mathcal{X}$, that is Q^* is LC.

The previous result can be stated more generally as follows:

Corollary 3. *We assume that for each $a \in \mathcal{A}$, \mathcal{P}_s^a is (weakly) continuous as a function of s , that is, if s_n converges to $s \in \mathcal{S}$ then for every bounded continuous function $f : \mathcal{S} \rightarrow \mathbb{R}$, $\int f d\mathcal{P}_{s_n}^a$ tends to $\int f d\mathcal{P}_s^a$. If $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is LC then $Q^* : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is LC.*

The proof of this result is similar as above but does not involve the Dominated Convergence Theorem as the continuity of $\mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)} [\max_{a' \in \mathcal{A}} Q_n(s', a')]$ as a function of s is ensured by the assumption of weakly continuity of \mathcal{P}_s^a .

We note that the assumption of weakly continuity of \mathcal{P}_s^a is weaker than the one on the existence of a density p_s^a as above. Indeed, if $s \mapsto p_s^a$ is continuous, then $s \mapsto \mathcal{P}_s^a$ is weakly continuous by Scheffé's lemma.

For the ULC case, the conditions and proof under which the implication " $\mathcal{P} + \mathcal{R} \implies Q^*$ " hold are stated by in Gelada et al. [2019]:

Corollary 4 (Gelada et al., 2019). *If \mathcal{R} and \mathcal{P} are ULC with Lipschitz constant $L_{\mathcal{R}}$ and $L_{\mathcal{P}}$ and $\gamma L_{\mathcal{P}} < 1$, then Q^* is ULC with Lipschitz constant $\frac{L_{\mathcal{R}}}{1 - \gamma L_{\mathcal{P}}}$.*

The reverse implication " $Q^* \implies \mathcal{R} + \mathcal{P}$ " is not true as shows the following counter-example. Let's suppose $\mathcal{S} = \mathbb{R}$ and $\mathcal{A} = \{1, 2\}$. Let $\mathcal{R}(s, 1) = 1$ and let $\mathcal{R}(s, 2)$ be any discontinuous function, for instance:

$$\mathcal{R}^*(s, 2) = \begin{cases} 0 & \text{if } s \leq s_0, s_0 \in \mathbb{R} \\ 0.5 & \text{if } s > s_0. \end{cases} \quad (3.8)$$

This leads to $Q^*(s, a) = \frac{1}{1 - \gamma}$ which is continuous but $\mathcal{R}(s, 2)$ is discontinuous.

Lemma E. *We assume that the next state probability measure \mathcal{P}_s^a admits a density $p_s^a : \mathcal{S} \rightarrow [0, \infty)$ with respect to the Lebesgue measure. If $s' \mapsto p_s^a(s')$ is bounded and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ and $s \mapsto p_s^a(s')$ are LC, then $Q^\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is LC.*

Similarly, we proceed by induction and consider the following sequence:

$$Q_0^\pi(s, a) = 0$$

$$Q_{n+1}^\pi(s, a) = \mathcal{R}(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}_s^a} V_n^\pi(S')$$

As above, Q_0^π is continuous and we then proceed by induction and apply the dominated convergence theorem.

We assume that Q_n^π is continuous. Assuming π -continuous, we have shown that this also implies that V_n^π is continuous.

1. $s \mapsto p_s^a(s')$ being continuous, s_m tends to $s \in \mathcal{S}$ implies that $p_{s_m}^a(s')V_n^\pi(s')$ tends to $p_s^a(s')V_n^\pi(s')$.
2. For all $s' \in \mathcal{S}$, $|V_n^\pi(s')| < V_{max}$ by assumption.

We fix $a \in \mathcal{A}$. p_s^a is bounded so there exists a function $h_a \in L^1(\mathcal{S})$ such that $p_s^a(s') \leq h_a(s')$ for all s', s .

By the dominated convergence theorem,

$$\lim_{m \rightarrow \infty} \int_{\mathcal{S}} p_{s_m}^a(s') V_n^\pi(s') ds' = \int_{\mathcal{S}} p_s^a(s') V_n^\pi(s') ds'.$$

Hence, if $s_m \rightarrow s$, then $\mathbb{E}_{s' \sim \mathcal{P}(\cdot | s_m, a)} V_n^\pi(s') \rightarrow \mathbb{E}_{s' \sim \mathcal{P}(\cdot | s, a)} V_n^\pi(s')$. The reward function being LC by assumption, Q_{n+1}^π is LC.

As shown above, Q_n^π converges to Q^π in sup norm, so Q^π is continuous.

Corollary 5. *We assume that for each $a \in \mathcal{A}$, \mathcal{P}_s^a is (weakly) continuous as a function of s , that is, if s_n converges to $s \in \mathcal{S}$ then for every bounded continuous function $f : \mathcal{S} \rightarrow \mathbb{R}$, $\int f d\mathcal{P}_{s_n}^a$ tends to $\int f d\mathcal{P}_s^a$. If $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ and $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ are LC then $Q^\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is LC.*

The proof of this result is similar as above but does not involve the Dominated Convergence Theorem as the continuity of $\mathbb{E}_{S' \sim \mathcal{P}(\cdot|s,a)} V^\pi(S')$ as a function of s is ensured by the assumption of weakly continuity of \mathcal{P}_s^a .

For the ULC case, the conditions and proof under which the implication " $\mathcal{P} + \mathcal{R} \implies Q^\pi$ " hold are stated by in Rachelson and Lagoudakis [2010]:

Lemma F (Rachelson and Lagoudakis, 2010). *If \mathcal{R} , \mathcal{P} and π are ULC with Lipschitz constant $L_{\mathcal{R}}$, $L_{\mathcal{P}}$ and L_π , and if $\gamma L_{\mathcal{P}}(1 + L_\pi) < 1$, then Q^π is ULC with Lipschitz constant $\frac{L_{\mathcal{R}}}{1 - \gamma L_{\mathcal{P}}(1 + L_\pi)}$.*

We note that the reverse implication " $Q^\pi \implies \mathcal{R} + \mathcal{P}$ " is not true as there exists a class of policies (the optimal policy is one element of this class) for which the implication does not hold.

3.B Proofs for Section 3.5

Lemma 3 (Identity metric). *$e^{\mathbb{I}}$ induces the finest topology on \mathcal{S} , made of all possible subsets of \mathcal{S} . Let (Y, d_Y) be any metric space. Any function h (resp. Any bounded h) : $(\mathcal{S}, e^{\mathbb{I}}) \rightarrow (Y, d_Y)$ is LC and UC (resp. ULC).*

Proof. Recall that for any (pseudo-)metric space (X, d) , a set $U \subseteq X$ is open if for any $x \in U$, there exists $r > 0$ such that the open ball $B_d(x, r)$ of radius r centered at x is a subset of U .

Suppose $U \subseteq \mathcal{S}$ is a non-empty open set of \mathcal{S} . Then for any $x \in U$, there exists $r > 0$ such that $B_{e^{\mathbb{I}}}(x, r) = \{y \in \mathcal{S} | e^{\mathbb{I}}(x, y) < r\} \subseteq U$. If $r > 1$, $B_{e^{\mathbb{I}}}(x, r) = \mathcal{S}$ and $U = \mathcal{S}$. Hence, $\mathcal{S} \subset \mathcal{T}_{e^{\mathbb{I}}}$. Else, $B_{e^{\mathbb{I}}}(x, r) = \{x\}$ and $\{x\} \subset U$. This is true for all $x \in U$ so $\cup_{x \in U} \{x\} \subset \mathcal{T}_{e^{\mathbb{I}}}$. Hence, $\mathcal{T}_{e^{\mathbb{I}}}$ is the collection of all open subsets of \mathcal{S} , that is, it is the discrete topology on \mathcal{S} .

Let (Y, d_Y) be any metric space.

- We first show that any function $h : (\mathcal{S}, e^{\mathbb{I}}) \rightarrow (Y, d_Y)$ is LC and UC.

Let $\epsilon > 0$. We choose $\delta = \frac{1}{2}$. Then, for all $x, y \in \mathcal{S}$,

$$\begin{aligned} e^{\mathbb{I}}(x, y) < \delta &\implies e^{\mathbb{I}}(x, y) = 0 \text{ as } e^{\mathbb{I}} : \mathcal{S} \times \mathcal{S} \rightarrow \{0, 1\}. \\ &\implies x = y \text{ as } e^{\mathbb{I}} \text{ is a proper metric.} \\ &\implies d_Y(h(x), h(y)) = 0 \text{ as } d_Y \text{ is a pseudometric.} \\ &\implies d_Y(h(x), h(y)) < \epsilon. \end{aligned}$$

This shows that any h is UC and thus LC.

- We now show that any bounded function $h : (\mathcal{S}, e^{\mathbb{I}}) \rightarrow (Y, d_Y)$ is ULC.

h is ULC if there exist $K > 0$ such that for all $x, x' \in X$ we have

$$d_Y(h(x), h(x')) \leq K d_{\mathcal{S}}(s, s').$$

If $s = s'$, then $d_{\mathcal{S}}(s, s') = 0$ by definition of $e^{\mathbb{I}}$ and $h(s) = h(s')$ which implies $d_Y(h(s), h(s')) = 0$. Hence, h is ULC.

Else, $e^{\mathbb{I}}(s, s') = 1$. h is bounded so $h(\mathcal{S})$ is a bounded subset of Y , that is for all $s, s' \in \mathcal{S}$, $d_Y(h(s), h(s')) \leq c$ for some $c > 0$. Hence, h is Lipschitz.

□

Lemma 4 (Trivial metric). $e^{\mathbb{T}}$ induces the coarsest topology on \mathcal{S} , consisting solely of $\{\emptyset, \mathcal{S}\}$. Let (Y, d_Y) be any metric space. Any function $h : (\mathcal{S}, e^{\mathbb{T}}) \rightarrow (Y, d_Y)$ is LC, UC and ULC iff h is constant.

Proof. For $e^{\mathbb{T}}$, suppose $U \subseteq \mathcal{S}$ is a non-empty open set of \mathcal{S} . Then for any $x \in U$, there exists $r > 0$ such that $B_{e^{\mathbb{T}}}(x, r) \subseteq U$. But observe that, for all $r > 0, x \in \mathcal{S}$, we have $B_{e^{\mathbb{T}}}(x, r) = \mathcal{S}$ by definition of the trivial pseudo-metric $e^{\mathbb{T}}$. Hence $U = \mathcal{S}$ and $(\mathcal{S}, e^{\mathbb{T}})$ has the trivial topology.

Let (Y, d_Y) be any metric space. Any function $h : (\mathcal{S}, e^{\mathbb{T}}) \rightarrow (Y, d_Y)$ is LC (resp UC, resp ULC) iff h is constant.

(\Leftarrow) : Suppose h is constant taking value $y \in Y$. Recall that $e^{\mathbb{T}} : \mathcal{S} \times \mathcal{S} \rightarrow \{0, 1\}$. It is clear that h must be Lipschitz continuous, and thus uniformly and locally continuous, because any $K > 0$ satisfies for all $s, s' \in \mathcal{S}$, $d_Y(h(s), h(s')) = d_Y(y, y) =$

$$0 \leq K e_t(x, y) = 0$$

(\implies) : Suppose for sake of contradiction that $h : \mathcal{S} \rightarrow Y$ is LC but not constant. Then there exist $s_1, s_2 \in \mathcal{S}$ such that $h(s_1) \neq h(s_2)$. This means that there exists $\epsilon_0 > 0$ such that $d_Y(h(x) - h(y)) > \epsilon_0$. Because we are using the trivial metric on \mathcal{S} , $d(x, y) = 0 < \delta$ for all $\delta > 0$. This contradicts the LC assumption of h . Hence, h LC implies that h is constant. This reasoning also holds for UC (resp ULC) as a function that cannot be LC cannot be UC (resp. ULC) (since $\text{ULC} \implies \text{UC} \implies \text{LC}$). \square

Lemma 5. *If $\eta = 0$, then any function f (resp. bounded function f): $(\mathcal{S}, d_{\mathcal{S}}) \rightarrow (Y, d_Y)$ is LC and UC (resp. ULC) with respect to the pseudometric $e^{\phi_{f,\eta}}$. However, given a function f and $\eta > 0$, there exists an η -abstraction $\phi_{f,\eta}$ such that f is not continuous with respect to $e^{\phi_{f,\eta}}$.*

As a byproduct, we can note that the metric e^{ϕ} induces the finest topology on $\hat{\mathcal{S}}$. Indeed, by definition, $e_{\phi} : \mathcal{S} \rightarrow \{0, 1\}$ is equal to the discrete pseudometric $e^{\mathbb{1}} : \hat{\mathcal{S}} \rightarrow \{0, 1\}$. Thanks to Lemma 3, we can deduce that e_{ϕ} induces the discrete topology on $\hat{\mathcal{S}}$.

Proof. • We first show that any function $f : (\mathcal{S}, d_{\mathcal{S}}) \rightarrow (Y, d_Y)$ is UC (and thus LC) with respect to the metric $e^{\phi_{f,0}}$.

Let $\epsilon > 0$. We choose $\delta = \frac{1}{2}$. Then, for all $s, t \in \mathcal{S}$, $e^{\phi_{f,0}}(s, t) < \delta \implies e^{\phi_{f,0}}(s, t) = 0 \implies \phi(s) = \phi(t) \implies f(s) = f(t) \implies |f(s) - f(t)| \leq \epsilon$.

- Then, the fact that any function $f : (\mathcal{S}, d_{\mathcal{S}}) \rightarrow (Y, d_Y)$ is ULC with respect to the pseudometric $e^{\phi_{f,0}}$ iff f is bounded is a consequence of Lemma 3
- Finally, if $\eta > 0$, there is no continuity guarantee about $f : (\mathcal{S}, d_{\mathcal{S}}) \rightarrow (Y, d_Y)$ with respect to the metric $e^{\phi_{f,\eta}}$.

Let $\epsilon > 0$. We choose $\delta = \frac{1}{2}$. Then, $e^{\phi_{f,\eta}}(s, t) < \delta \implies e^{\phi_{f,\eta}}(s, t) = 0 \implies \phi(s) = \phi(s') \implies |f(s) - f(s')| \leq \eta$.

If $\epsilon \geq \eta$, then the continuity definition is respected.

But when $\epsilon < \eta$, we cannot conclude anything.

□

Lemma 6. Q^* (resp. Q^π) is ULC with Lipschitz constant 1 with respect to d^\sim (resp. $d^{\sim\pi}$).

Proof. Take any $s, t \in \mathcal{S}$ and $a \in \mathcal{A}$.

$$\begin{aligned}
|Q^*(s, a) - Q^*(t, a)| &= \left| \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_s^a(s') V^*(s') - \left(\mathcal{R}_t^a + \gamma \sum_{t' \in \mathcal{S}} \mathcal{P}_t^a(t') V^*(t') \right) \right| \\
&= \left| \mathcal{R}_s^a - \mathcal{R}_t^a + \gamma \sum_{s' \in \mathcal{S}} V^*(s') (\mathcal{P}_s^a(s') - \mathcal{P}_t^a(s')) \right| \\
&\leq |\mathcal{R}_s^a - \mathcal{R}_t^a| + \gamma \left| \sum_{s' \in \mathcal{S}} V^*(s') (\mathcal{P}_s^a(s') - \mathcal{P}_t^a(s')) \right| \\
&\leq |\mathcal{R}_s^a - \mathcal{R}_t^a| + \gamma \mathcal{W}(d^\sim)(\mathcal{P}_s^a(s'), \mathcal{P}_t^a(s')) \\
&\leq \max_{a \in \mathcal{A}} \{ |\mathcal{R}_s^a - \mathcal{R}_t^a| + \gamma \mathcal{W}(d^\sim)(\mathcal{P}_s^a(s'), \mathcal{P}_t^a(s')) \} \\
&= d^\sim(s, t)
\end{aligned}$$

We can show the result for Q^π similarly. □

Corollary 1. Q^* (resp. Q^π) is ULC with Lipschitz constant V_{\max} with respect to e^\sim (resp. $e^{\sim\pi}$).

Proof. This is a consequence of Lemma 6, the normalization coming from the fact that the bisimulation metric d^\sim is bounded by $\frac{R_{\max}}{1-\gamma}$. □

Lemma 7. For a given MDP, let Q^π be the Q -function of policy π , and Q^* the optimal Q -function. The following are continuous pseudo-metrics:

1. $d_{\Delta^*}(s, s') = \max_{a \in \mathcal{A}} |Q^*(s, a) - Q^*(s', a)|$
2. $d_{\Delta^\pi}(s, s') = \max_{a \in \mathcal{A}} |Q^\pi(s, a) - Q^\pi(s', a)|$
3. $d_{\Delta^\forall}(s, s') = \max_{\pi \in \Pi, a \in \mathcal{A}} |Q^\pi(s, a) - Q^\pi(s', a)|$

Q^* (resp. Q^π) is ULC with Lipschitz constant 1 wrt to d_{Δ^*} (resp. d_{Δ^π}). Q^π is ULC with Lipschitz constant 1 wrt to d_{Δ^\forall} for any $\pi \in \Pi$.

Proof. Let's show that these functions are pseudometrics.

First, $d_{\Delta^*}(s, s) = 0$. Second, $d_{\Delta^*}(s, s') = d_{\Delta^*}(s', s)$ by symmetry of the graph of the absolute value function. Finally,

$$\begin{aligned} d_{\Delta^*}(s_1, s_2) &= \max_{\mathcal{A}} |Q^*(s_1, a) - Q^*(s_3, a) + Q^*(s_3, a) - Q^*(s_2, a)| \\ &\leq \max_{\mathcal{A}} (|Q^*(s_1, a) - Q^*(s_3, a)| + |Q^*(s_3, a) - Q^*(s_2, a)|) \\ &\leq \max_{\mathcal{A}} |Q^*(s_1, a) - Q^*(s_3, a)| + \max_{\mathcal{A}} |Q^*(s_3, a) - Q^*(s_2, a)| \\ &= d_{\Delta^*}(s_1, s_3) + d_{\Delta^*}(s_3, s_2), \end{aligned}$$

where in the second line we apply the triangle inequality. Hence, d_{Δ^*} satisfies the triangle inequality. We can thus conclude that d_{Δ^*} is a pseudometric. Similarly, we prove that d_{Δ^π} and d_{Δ^\vee} are pseudometrics.

We now prove the continuity properties given by these three metrics.

- Let $s, t \in \mathcal{S}$. As above, we fix $a \in \mathcal{A}$. Then, $|Q^*(s, a) - Q^*(t, a)| \leq \max_{a \in \mathcal{A}} |Q^*(s, a) - Q^*(t, a)| = d_{\Delta^*}(s, t)$. Thus, Q^* is Lipschitz continuous with respect to d_{Δ^*} .
- Let $s, t \in \mathcal{S}$ and let $\pi \in \Pi$. As before, let's fix $a \in \mathcal{A}$. $|Q^\pi(s, a) - Q^\pi(t, a)| \leq \max_{a \in \mathcal{A}} |Q^\pi(s, a) - Q^\pi(t, a)| = d_{\Delta^\pi}(s, t)$. Thus, Q^π (resp. Q^*) is Lipschitz continuous with respect to d_{Δ^π} (resp. d_{Δ^*}).
- Let $s, t \in \mathcal{S}$ and let $\pi \in \Pi$. As before, let's fix $a \in \mathcal{A}$. $|Q^\pi(s, a) - Q^\pi(t, a)| \leq \max_{\pi \in \Pi} \max_{a \in \mathcal{A}} |Q^\pi(s, a) - Q^\pi(t, a)| = d_{\Delta^\vee}(s, t)$. This in particular true for any $\pi \in \Pi$, hence for any policy π , Q^π is Lipschitz continuous with respect to d_{Δ^\vee} .

□

We now formalize in Lemma G and Lemma H the results from Taylor et al. [2009] that we added to Table 3.2.

Lemma G. $\forall s, s' \in \mathcal{S}, |V^*(s) - V^*(s')| \leq d^{\sim \text{tax}}(s, s')$

Proof. The proof of this result can be found in [Taylor et al., 2009].

□

Lemma H. $\forall s, s' \in \mathcal{S}, \frac{1-\gamma}{R_{\max}} |V^*(s) - V^*(s')| \leq e^{\sim lax}(s, s')$

Proof. This result is a consequence of Lemma G. The normalization comes from the fact that the lax bisimulation metric $d^{\sim lax}$ is bounded by $\frac{R_{\max}}{1-\gamma}$. \square

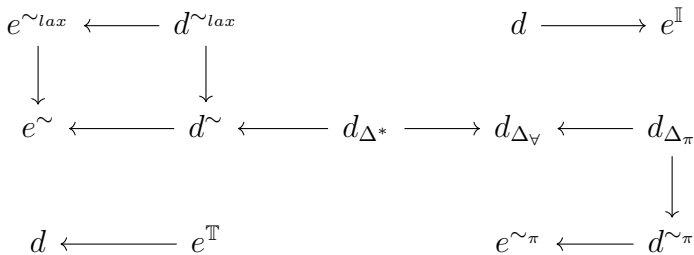
Remark. When \mathcal{S} is finite, the number of policies to consider to compute d_{Δ_v} is finite: $d_{\Delta_v}(s, s') = \max_{\pi \in \Pi, a \in \mathcal{A}} |Q^\pi(s, a) - Q^\pi(s', a)| = \max_{\pi \in \Pi_{AVF}, a \in \mathcal{A}} |Q^\pi(s, a) - Q^\pi(s', a)|$, where Π_{AVF} is the finite set of extremal policies corresponding to Adversarial Value Functions (AVFs) [Bellemare et al., 2019].

Proof. The space of value functions $\{V^\pi | \pi \in \Pi\}$ is a polytope [Dadashi et al., 2019] and Bellemare et al. [2019] considered the finite set of policies Π_{AVF} corresponding to extremal vertices of this polytope $\{V^\pi | \pi \in \Pi_{AVF}\}$

As noted by Dabney et al. [2021], the space of action value functions $\{Q^\pi | \pi \in \Pi\}$ is also polytope since polytopes are invariant by translations (reward \mathcal{R}_s^a term) and linear transformations ($\gamma \mathcal{T}_s^a$ term). Additionally, extremal vertices of $\{Q^\pi | \pi \in \Pi\}$ and $\{V^\pi | \pi \in \Pi\}$ are reached for the same policies as extremal points of a polytope are invariant by affine transformations. Hence, the set of extremal vertices of $\{Q^\pi | \pi \in \Pi\}$ is $\{Q^\pi | \pi \in \Pi_{AVF}\}$. The maximum between two elements of this polytope is reached at two extremal vertices of the polytope, hence the result.

Now, when approximating the metric d_{Δ_v} by $d_{\widetilde{AVF}(n)} = \max_{\pi \in \Pi_{\widetilde{AVF}(n)}, a \in \mathcal{A}} |Q^\pi(s, a) - Q^\pi(s', a)|$, where $\Pi_{\widetilde{AVF}(n)}$ are n samples from the set of extremal policies Π_{AVF} , it follows that $d_{\widetilde{AVF}(n)} \leq d_{\Delta_v}$. \square

Theorem 2. The relationships between the topologies induced by the metrics in Table 3.2 are given by the following diagram. We denote by $d_1 \rightarrow d_2$ when $\mathcal{T}_{d_1} \subset \mathcal{T}_{d_2}$, that is, when \mathcal{T}_{d_1} is coarser than \mathcal{T}_{d_2} . Here d denotes any arbitrary metric.



We now prove the second theorem of our paper, Theorem 2. The proof itself will be made up of a series of lemmas for each of the arrows in the diagram. We first start by proving a necessary lemma.

Lemma I. *Given two metrics d_1 and d_2 on \mathcal{S} , if there exists $\alpha > 0$ such that $d_1(s, t) \leq \alpha d_2(s, t)$ for all $s, t \in \mathcal{S}$, then \mathcal{T}_{d_1} is coarser than \mathcal{T}_{d_2} , that is $\mathcal{T}_{d_1} \subset \mathcal{T}_{d_2}$.*

Proof. Let $\epsilon > 0$ and $x \in \mathcal{S}$. Suppose $x' \in B_{d_2}(x, \epsilon)$. By definition, this means that $d_2(x, x') \leq \epsilon$. It implies $\frac{1}{\alpha}d_1(x, x') \leq d_2(x, x') \leq \epsilon$ by assumption and then $x' \in B_{d_1}(x, \alpha\epsilon)$. Hence, $B_{d_2}(x, \epsilon) \subset B_{d_1}(x, \alpha\epsilon)$.

Now, suppose $U \subset \mathcal{S}$ is a non-empty open set of \mathcal{S} . Then $\forall x \in U$, there exists $r > 0$ such that $B_{d_1}(x, r) \subset U$. We have shown that $B_{d_2}(x, \epsilon) \subset B_{d_1}(x, \alpha\epsilon)$ for all $\epsilon > 0$. So we also have $B_{d_2}(x, \frac{r}{\alpha}) \subset U$. By definition a topology on \mathcal{S} is a collection of open subsets on \mathcal{S} so we can conclude that $\mathcal{T}_{d_1} \subset \mathcal{T}_{d_2}$. \square

Lemma J. *For all $s, t \in \mathcal{S}$, $e^{\mathbb{T}}(s, t) \leq e^{\mathbb{I}}(s, t)$ and $\mathcal{T}_{e^{\mathbb{T}}} \subset \mathcal{T}_{e^{\mathbb{I}}}$.*

Proof. This comes directly from the definitions of the trivial metric $e^{\mathbb{T}}$ and discrete metric $e^{\mathbb{I}}$. Moreover, as mentioned in Lemma 3 and Lemma 4, the discrete metric induces the finest topology on \mathcal{S} while the trivial metric induces the coarsest topology. \square

Lemma K. *For all $s, t \in \mathcal{S}$, $d_{\Delta^*}(s, t) \leq d_{\Delta}(s, t)$ and $d_{\Delta^{\pi}}(s, t) \leq d_{\Delta}(s, t)$.*

Proof. By definition, for all $s, s' \in \mathcal{S}$, $d_{\Delta^*}(s, s') = \max_{a \in \mathcal{A}} |Q^*(s, a) - Q^*(s', a)| \leq \max_{\pi \in \Pi, a \in \mathcal{A}} |Q^{\pi}(s, a) - Q^{\pi}(s', a)|$ so $d_{\Delta^*}(s, t) \leq d_{\Delta}(s, t)$.

The proof is similar for $d_{\Delta^{\pi}}$ \square

Lemma L. *For all $s, t \in \mathcal{S}$, there exists $\alpha > 0$ such that $d^{\sim}(s, t) \leq \alpha e^{\sim}(s, t)$ and $d^{\sim lax}(s, t) \leq \alpha e^{\sim lax}(s, t)$ and $d^{\sim \pi}(s, t) \leq \alpha e^{\sim \pi}(s, t)$.*

Proof. Let $s, t \in \mathcal{S}$.

If $e^{\sim}(s, t) = 0$, then $s \sim t$ and $d^{\sim}(s, t) = 0$.

If $e^{\sim}(s, t) = 1$, then $d^{\sim}(s, t) \neq 0$. Moreover, by construction of the bisimulation metric, $d^{\sim}(s, t) \in [0, \frac{R_{\max}}{1-\gamma}]$ for all $s, t \in \mathcal{S}$. Hence, $d^{\sim}(s, t) \leq \frac{R_{\max}}{1-\gamma} e^{\sim}(s, t)$. Choosing

$\alpha = \frac{R_{\max}}{1-\gamma}$, we get $d^\sim(s, t) \leq \alpha e^\sim(s, t)$.

The proof is similar for the two other inequalities. \square

Lemma M. *For all $s, t \in \mathcal{S}$, $e^{\sim_{\text{Iax}}}(s, t) \leq e^\sim(s, t)$ and $d^{\sim_{\text{Iax}}}(s, t) \leq d^\sim(s, t)$.*

Proof. Let $s, t \in \mathcal{S}$.

If $e^{\sim_{\text{Iax}}}(s, t) = 0$ then the inequality is verified by postivity of any metric.

If $e^{\sim_{\text{Iax}}}(s, t) = 1$, it means there exists $a \in \mathcal{A}$ such that for all $b \in \mathcal{A}$, $\mathcal{R}_s^a \neq \mathcal{R}_s^b$ and there exists $X \in \Sigma(E)$ such that $\mathcal{P}_s^a(X) \neq \mathcal{P}_b^a(X)$. This is in particular true when $b = a$ which means that the conditions to be a bisimulation relation are not satisfied. Hence, $e^\sim(s, t) = 1$.

The second inequality is proven in Taylor et al. [2009]. \square

Complexity results

We now provide details explaining the complexity results from Table 3.2.

The discrete identity metric $e^\mathbb{I}$ compares all pairs of states which results in a complexity $O(|\mathcal{S}|)$. The complexity of the trivial metric $e^\mathbb{T}$ is independent of the number of states and hence constant.

The discrete bisimulation metric can be computed by finding the bisimulation equivalence classes. This can be done by starting with a single equivalence class that gets iteratively split into smaller equivalence classes when one of the bisimulation conditions is violated; this process is repeated until stability. Each iteration of this process is $O(|\mathcal{A}||\mathcal{S}|^2)$, since we are performing an update for all actions and pairs of states. Since there can be at most $|\mathcal{S}|$ splits, this yields the complexity of $O(|\mathcal{A}||\mathcal{S}|^3)$.

The bisimulation metric can be computed by iteratively applying $\lfloor \frac{\ln \delta}{\ln \gamma} \rfloor$ times the operator F from Lemma 1 [Ferns et al., 2004], for each action and each pair of states. We note that the time complexity for solving an optimal flow problem as originally presented by Ferns et al. [2004] is incorrect and off by a factor of $|\mathcal{S}|$: it should be $O(|\mathcal{S}|^3 \log |\mathcal{S}|)$. Thus, the complexity they present for computing the bisimulation metric is also off by a factor of $|\mathcal{S}|$; the corrected time complexity is $O(|\mathcal{A}||\mathcal{S}|^5 \log |\mathcal{S}| \frac{\ln \delta}{\ln \gamma})$, as we presented in Table 3.1. The π -bisimulation metrics do

not require a loop over the action space as the matching is under a fixed policy. Therefore their complexity is the one of the bisimulation metrics off by a factor $|\mathcal{A}|$.

The time complexity of computing d_{Δ^π} and d_{Δ^*} is the same as the complexity of policy evaluation and value iteration, plus an extra $O(|\mathcal{S}^2|)$ term for computing the resulting metric; this last term is dominated by the policy/value iteration complexity, which is what we have included in the table.

3.C Formal Definition of Bisimulation Metrics

We will also define a metric between probability functions that is used by some of the state metrics considered in this paper. Let (Y, d_Y) be a metric space with Borel σ -algebra Σ .

Definition 6. The **Wasserstein** distance [Villani, 2008] between two probability measures P and Q on Y , under a given metric d_Y is given by $W_{d_Y}(P, Q) = \inf_{\lambda \in \Gamma(P, Q)} \mathbb{E}_{(x, y) \sim \lambda} [d_Y(x, y)]$, where $\Gamma(P, Q)$ is the set of couplings between P and Q .

The Wasserstein distance can be understood as the minimum cost of transporting P into Q where the cost of moving a unit mass from the point x to the point y is given by $d(x, y)$.

Theorem 4. Define $\mathcal{F}^\pi : \mathcal{M} \rightarrow \mathcal{M}$ by $\mathcal{F}^\pi(d)(s, t) = |\mathcal{R}_s^\pi - \mathcal{R}_t^\pi| + \gamma \mathcal{W}_1(d)(\mathcal{P}_s^\pi, \mathcal{P}_t^\pi)$, then \mathcal{F}^π has a least fixed point d_\sim^π , and d_\sim^π is a π -bisimulation metric.

Definition 7. Given a 1-bounded pseudometric $d \in \mathbb{M}$, the metric $\delta(d) : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ is defined as follows:

$$\delta(d)((s, a), (t, b)) = |\mathcal{R}(s, a) - \mathcal{R}(t, b)| + \gamma \mathcal{W}(d)(\mathcal{P}(s, a), \mathcal{P}(t, b))$$

Definition 8. Given a finite 1-bounded metric space (\mathfrak{M}, d) let $\mathcal{C}(\mathfrak{M})$ be the set of compact spaces (e.g. closed and bounded in \mathbb{R}). The Hausdorff metric $H(d) : \mathcal{C}(\mathfrak{M}) \times \mathcal{C}(\mathfrak{M}) \rightarrow [0, 1]$ is defined as:

$$H(d)(X, Y) = \max \left(\sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y) \right)$$

Definition 9. Denote $X_s = \{(s, a) | a \in \mathcal{A}\}$. We define the operator $F : \mathbb{M} \rightarrow \mathbb{M}$ as:

$$F(d)(s, t) = H(\delta(d))(X_s, X_t)$$

Theorem 5. F is monotonic and has a least fixed point $d^{\sim lax}$ in which $d^{\sim lax}(s, t) = 0$ iff $s \sim_{lax} t$.

3.D Additional Empirical Evaluations

In this section we conduct an empirical evaluation to complement the theoretical analyses performed above. We conduct these experiments on the well-known Four Rooms domain [Sutton et al., 1999, Solway et al., 2014, Machado et al., 2017, Bellemare et al., 2019] for all our experiments, which is illustrated in Figure 3.3. This domain enables clear visualization and ensures that we can compute a metric defined over the entirety of the state space. These experiments aim to showcase 1) the qualitative difference in the state-wise distances produced by the different metrics; 2) visualize the differences in abstract states that the different metrics produce when used for state aggregation. Results are showcased in Figure 3.4 and Figure 3.5.

The dynamics of the environment are as follows. There are four actions (up, down, left, right), transitions are deterministic, there is a reward of +1 upon entering the non-absorbing goal state, there is a penalty of -1 for running into a wall, and we use a discount factor $\gamma = 0.9$. We will conduct our experiments on four representative metrics: d^{\sim} , $d^{\sim lax}$, d_{Δ^*} , and d_{Δ} . Since the maximization over all policies required for d_{Δ} is in general intractable, we instead sample 50 adversarial value functions (AVFs) [Bellemare et al., 2019] as a proxy for the set of all policies.

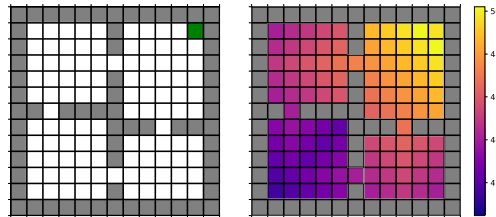


Figure 3.3: Four Rooms domain with a single goal state in green (left). Optimal values for each cell (right).

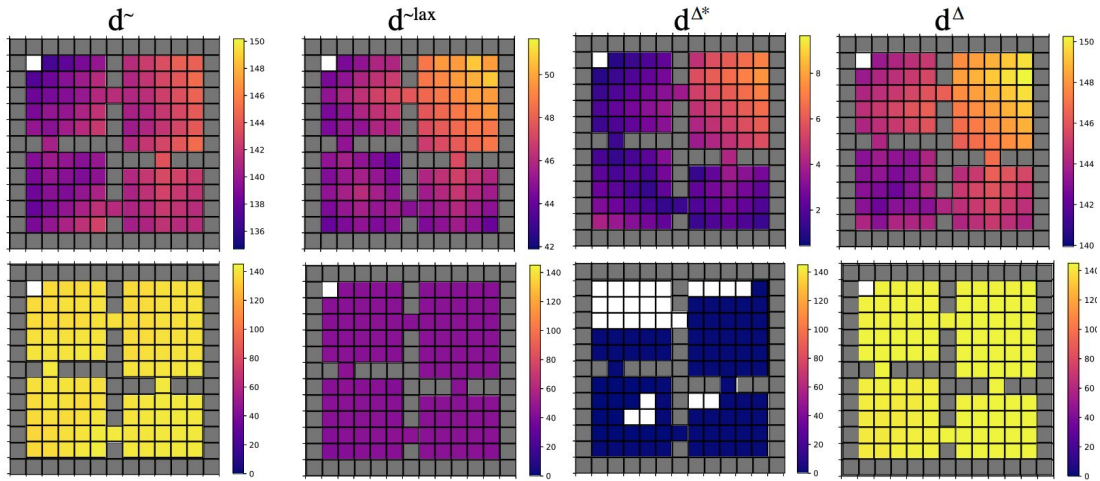


Figure 3.4: The top row illustrates the distances from the top-left cell to every other cell (note the color scales are shifted for each metric for easier differentiation between states). The bottom row displays $d(s, t) - |V^*(s) - V^*(t)|$, where s is the top-left cell, illustrating how tight an upper bound the metrics yield on the difference in optimal values.

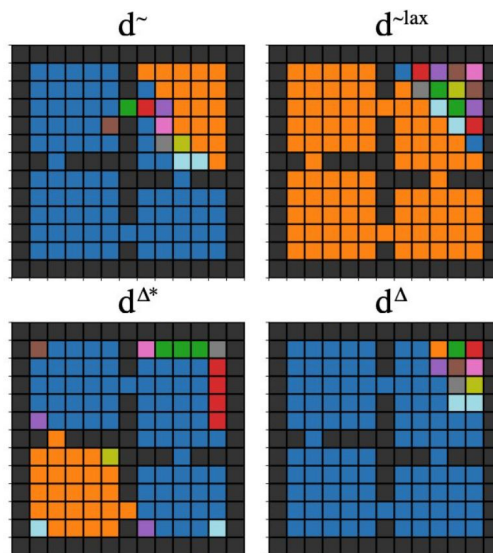


Figure 3.5: State clusters produced by the different metrics when targeting 11 aggregate states. There is no color correlation across metrics.


Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor (**only required where there isn't already a statement of contribution within the paper itself**).


Title of Paper	Metrics and continuity in reinforcement learning
Publication Status	<input checked="" type="checkbox"/> Published <input type="checkbox"/> Accepted for Publication <input type="checkbox"/> Submitted for Publication <input type="checkbox"/> Unpublished and unsubmitted work written in a manuscript style
Publication Details	Charline Le Lan, Marc G. Bellemare, Pablo Samuel Castro. Metrics and continuity in reinforcement learning. <i>In AAAI Conference on Artificial Intelligence (AAAI) 2021.</i>

Student Confirmation

Student Name:	Charline Le Lan		
Contribution to the Paper	<p>I led the project, proved most theoretical results, wrote the first version of the codebase and paper, ran experiments and generated the plots included in the paper. Marc suggested to compare different metrics through the lens of continuity. Pablo wrote a second version of the codebase building on my implementation.</p> <p>Marc and Pablo advised the project, provided feedback and edits on the paper.</p>		
Signature		Date	March 23, 2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Marc G. Bellemare			
Supervisor comments An excellent piece of work that set the stage for further discoveries.			
Signature		Date	13/04/23

This completed form should be included in the thesis, at the end of the relevant chapter.

4

On the Generalization of Representations in Reinforcement Learning

Abstract

In reinforcement learning, state representations are used to tractably deal with large problem spaces. State representations serve both to approximate the value function with few parameters, but also to generalize to newly encountered states. Their features may be learned implicitly (as part of a neural network) or explicitly (for example, the successor representation of Dayan [1993]). While the approximation properties of representations are reasonably well-understood, a precise characterization of how and when these representations generalize is lacking. In this work, we address this gap and provide an informative bound on the generalization error arising from a specific state representation. This bound is based on the notion of effective dimension which measures the degree to which knowing the value at one state informs the value at other states. Our bound applies to any state representation and quantifies the natural tension between representations that generalize well and those that approximate well. We complement our theoretical results with an empirical survey of classic representation learning methods from the literature and results on the Arcade Learning Environment, and find that the generalization behaviour of learned representations is well-explained by their effective dimension.

4.1 Introduction

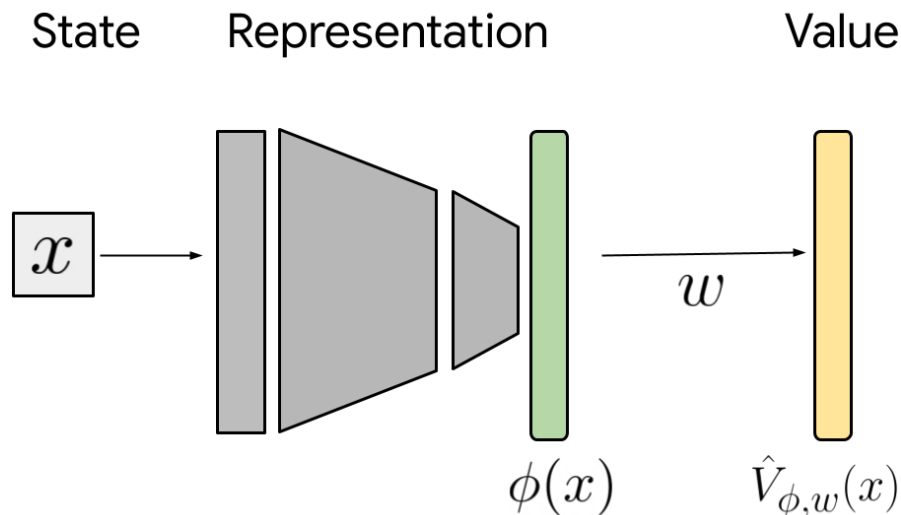


Figure 4.1: A deep RL architecture seen as a deep representation ϕ and a value prediction $\hat{V}_{\phi,w}$.

At the heart of reinforcement learning (RL) is the problem of predicting the expected return that can be obtained from different states. In most practical situations, these predictions are made on the basis of parametric function approximation, needed in order to make accurate predictions on the basis of limited samples – technically speaking, to estimate the *value function* [Sutton and Barto, 2018]. Linear function approximation, for example, estimates the value function using a fixed state representation ϕ which maps states to vectors in \mathbb{R}^k ; general-purpose algorithms for constructing state representations include tile coding [Sutton, 1996], the Fourier basis [Konidaris et al., 2011], local basis functions [Ratitch and Precup, 2004], and methods based on properties of the transition function [Mahadevan and Maggioni, 2007, Ghosh and Bellemare, 2020]. Common deep RL network architectures such as DQN [Mnih et al., 2015] use multiple layers of nonlinear transformations to map perceptual inputs to a final layer which is linearly transformed into a value function prediction (Figure 4.1); accordingly, we may also view this final layer as a (time-varying) state representation ϕ [Levine et al., 2017, Chung et al., 2018].

It is generally believed that auxiliary tasks, known to improve performance in deep reinforcement learning [Jaderberg et al., 2017, Bellemare et al., 2017],

play an important role in shaping the learned state representation [Bellemare et al., 2019, Dabney et al., 2021, Lyle et al., 2021]. This motivates the need to understand how representation learning impacts policy evaluation. In this paper, we give a theoretical characterization of the generalization properties of a given or learned representation. While there are a number of results characterizing the approximation error due to a representation [Petrik, 2007, Parr et al., 2008], its effect on statistical error is relatively unknown.

Our first contribution is a bound on the generalization error (approximation + estimation) that arises when performing Monte Carlo value function estimation with a given k -dimensional representation ϕ (Section 4.3). Critically, this bound depends on the (in)coherence of the feature matrix Φ [Candès and Recht, 2009], which in turns defines the *effective dimension* of the representation. This effective dimension determines how many samples are needed to obtain a good generalization of the value function with the chosen representation; it may be as low as k , indicating that generalization is as good as possible, or as high as $|S|$, the number of states, indicating no generalization at all. The bound applies more broadly to the generalization error incurred in least-squares regression problems where a subset of a larger set of points is observed.

In Section 4.4, we demonstrate the usefulness of our bound by specializing it to study the generalization properties of the successor representation (SR) [Dayan, 1993]. Specifically, we consider the state representation constructed from the top k singular vectors of the SR [Stachenfeld et al., 2014, Machado et al., 2017, Behzadian and Petrik, 2018]. Empirically, we find that the effective dimension of this representation – and consequently its generalization characteristics – can vary substantially according to the transition structure of the environment. We also show empirically that the effective dimension is important to determine the generalization capacity of different theoretically-motivated representations in the Four Rooms domain [Sutton et al., 1999].

In an empirical study on the Arcade Learning Environment [Bellemare et al., 2013], we find that the notions of incoherence and effective dimension correlate with

the observed empirical performance of existing value-based deep RL agents (Subsection 4.5.2). Furthermore, we find that a simple auxiliary loss motivated by our bound shows promising gains in the offline deep RL setting.

4.2 Background

We consider a Markov Decision Process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$ [Puterman, 1994] with finite state space \mathcal{S} , discrete set of actions \mathcal{A} , transition kernel $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{P}(\mathcal{S})$, deterministic reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow [-R_{\max}, R_{\max}]$, and discount factor $\gamma \in [0, 1)$. For simplicity, we make the correspondence $\mathcal{S} = \{1, \dots, S\}$. We write \mathcal{P}_s^a to denote the next-state distribution over \mathcal{S} resulting from selecting action a in s and write \mathcal{R}_s^a for the corresponding reward.

A stationary policy $\pi : \mathcal{S} \rightarrow \mathcal{P}(\mathcal{A})$ is a mapping from states to distributions over actions, describing a particular way of interacting with the environment. We denote the set of all policies by Π . For any policy $\pi \in \Pi$, the value function $V^\pi(s)$ measures the expected discounted sum of rewards received when starting from state $s \in \mathcal{S}$ and acting according to π :

$$V^\pi(s) := \mathbb{E}_{\pi, \mathcal{P}} \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}_{S_t}^{A_t} \mid S_0 = s, A_t \sim \pi(\cdot \mid S_t) \right].$$

The upper-bound value is $V_{\max} := \frac{R_{\max}}{1-\gamma}$. In vector notation [Puterman, 1994], let $r_\pi \in \mathbb{R}^S$ denote the vector of expected rewards, and let $P_\pi \in \mathbb{R}^{S \times S}$ be the transition matrix whose entries are

$$P_\pi(s, s') = \sum_{a \in \mathcal{A}} \mathcal{P}_s^a(s') \pi(a \mid s).$$

We then have

$$V^\pi = \sum_{t=0}^{\infty} (\gamma P_\pi)^t r_\pi = (I - \gamma P_\pi)^{-1} r_\pi.$$

In this paper we consider approximating the value function V^π using a linear combination of features. We call the map $\phi : \mathcal{S} \rightarrow \mathbb{R}^k$ a *k-dimensional state*

representation; $\phi(s)$ is the feature vector for a state $s \in \mathcal{S}$. In general, we will be interested in the setting where $k \ll S$. The value function approximation at s is

$$V_{\phi,w}(s) = \phi(s)^\top w,$$

where $w \in \mathbb{R}^k$ is a weight vector. We collect the per-state feature vectors into a feature matrix $\Phi \in \mathbb{R}^{S \times k}$. For simplicity, we assume Φ has full column rank. In vector form, the value function approximation (a S -dimensional vector) is more directly expressed as

$$V_{\phi,w} = \Phi w.$$

4.2.1 Statistical Learning Theory

We consider the *batch Monte Carlo policy evaluation* setting, in which we are given a sample of training examples $D = \{(s_1, y_1), \dots, (s_n, y_n)\} \in (\mathcal{S} \times \mathbb{R})^n$ and wish to determine a good linear approximation to V^π on the basis of this sample. Here, s_i is a state and y_i is a realisation of the random return $G^\pi(s_i)$ [Bellemare et al., 2017, Sutton and Barto, 2018], defined by the random-variable equation

$$G^\pi(s) = \sum_{t=0}^{\infty} \gamma^t \mathcal{R}_{s_t}^{a_t}, \quad s_0 = s, a_t \sim \pi(\cdot | s_t).$$

We assume that s_i is drawn uniformly at random from \mathcal{S} .¹ The batch Monte Carlo setting obviates some of the technical challenges in analyzing iterative methods such as least-squares TD (LSTD) but still allows us to provide practically-relevant theoretical guarantees.

We measure the quality of a linear approximation $V_{\phi,w}$ in terms of the expected squared error

$$R(V_{\phi,w}) = \frac{1}{S} \sum_{s \in \mathcal{S}} \mathbb{E}_{y \sim G^\pi(s)} (V_{\phi,w}(s) - y)^2. \quad (4.1)$$

For a value function V , we express this error and related quantities in terms of the uniformly-weighted L^2 norm

$$\|V\|_{\mathcal{S},2} = \sqrt{\frac{1}{S} \sum_{s \in \mathcal{S}} (V(s))^2}.$$

¹Results for a larger class of distributions are given in Appendix 4.A

Following terminology from statistical learning theory [Vapnik, 1995], we call $R(V_{\phi,w})$ the *population risk* of $V_{\phi,w}$. One can verify that $R(V_{\phi,w})$ is minimized when $V_{\phi,w} = V^\pi$.

Given the dataset D and a fixed state representation ϕ , least-squares regression determines the weight vector \hat{w} minimizing the *empirical risk function*

$$\hat{R}(V_{\phi,w}) = \frac{1}{n} \sum_{i=1}^n (V_{\phi,w}(s_i) - y_i)^2.$$

Notice that \hat{R} is a random function as it depends on the training sample D .

We are interested in the performance of the least-squares approximation $V_{\phi,\hat{w}}$ compared to the true value function V^π . Let us denote by V_{ϕ,w^*} the linear approximation minimizing the population risk, such that

$$w^* = \arg \min_{w \in \mathbb{R}^k} R(V_{\phi,w}).$$

For clarity of exposition, we will assume this approximation is unique. The *excess risk* $\mathcal{E}(V_{\phi,\hat{w}}) = R(V_{\phi,\hat{w}}) - R(V^\pi)$ measures the additional error suffered by the approximation $V_{\phi,\hat{w}}$ compared to the true value function. We decompose it into an estimation error term, measuring the performance gap with the best-in-class, and an approximation error term arising from considering a restricted set of k -dimensional value function approximations:

$$\mathcal{E}(V_{\phi,\hat{w}}) = \underbrace{R(V_{\phi,\hat{w}}) - R(V_{\phi,w^*})}_{\text{estimation error}} + \underbrace{R(V_{\phi,w^*}) - R(V^\pi)}_{\text{approximation error}}.$$

4.2.2 The Successor Representation

The successor representation [Dayan, 1993] describes a state in terms of the frequency at which it visits future states; it is also related to the fundamental matrix in the study of Markov chains see Kemeny and Snell [1961], Brémaud [2013], Grinstead and Snell [2012].

Definition 10. *The successor representation (SR) with respect to a policy π for a state $s \in \mathcal{S}$ is the expected discounted sum of future occupancies for each state $s' \in \mathcal{S}$. Specifically, $\psi^\pi(s) = (\psi^\pi(s, s'))_{s' \in \mathcal{S}}$, where*

$$\psi^\pi(s, s') = \mathbb{E}_{\pi, \mathcal{P}} \left[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}[s_t = s'] \mid s_0 = s \right].$$

Expressed as a matrix $\Psi^\pi \in \mathbb{R}^{S \times S}$, the successor representation can be written as:

$$\Psi^\pi = (I - \gamma P_\pi)^{-1}.$$

As a consequence of the Bellman equation, we can express the value function in terms of the successor representation as follows:

$$V^\pi = \Psi^\pi r_\pi.$$

This makes it a particularly appealing candidate to use as a state representation. In particular, it is well-established that the top eigenvectors [Mahadevan and Maggioni, 2007] or singular vectors [Behzadian and Petrik, 2018] of the successor representation form a useful representation [Stachenfeld et al., 2014]. Petrik [2007] derived an analytical bound on the approximation error for linear value function approximation for a representation made of the top eigenvectors of Ψ^π in the particular setting where P_π is symmetric. By contrast, in this paper, we consider the more general setting of an arbitrary transition matrix P_π and consider a generalization bound that accounts for the statistical nature of the learning process.

4.3 Characterizing Excess Risk

Our first result characterizes how the choice of representation affects the generalization of value functions. Theorem 6 applies beyond the setting of reinforcement learning, and more generally characterizes the excess risk of a broad class of least-squares regression problems.

To begin, we assume that the labels y_1, \dots, y_n satisfy

$$y_i = V(s_i) + \eta_i,$$

where $V : \mathcal{S} \rightarrow \mathbb{R}$ and η_i is i.i.d. zero mean σ -sub-Gaussian noise [Vershynin, 2010]. This includes the batch Monte Carlo setting, in which case $V = V^\pi$ and $\eta_i \stackrel{D}{=} G^\pi(s_i) - V^\pi(s_i)$, where $G^\pi(s_i)$ is the random return from s_i .

For a feature matrix Φ , we write P_Φ for the orthogonal projector onto its column space, and P_Φ^\perp for the orthogonal projector onto the corresponding nullspace. We have

$$P_\Phi = \Phi(\Phi^\top \Phi)^{-1} \Phi^\top \quad P_\Phi^\perp = I_S - P_\Phi.$$

In particular, the approximation error for a given state representation ϕ is

$$R(V_{\phi, w^*}) - R(V) = \|P_\Phi^\perp V\|_{S,2}^2.$$

A key quantity in our analysis is the notion of the *effective dimension* of a state representation, which dictates the number of samples required to achieve a low estimation error.

Definition 11 (Effective dimension). *Let $\Phi \in \mathbb{R}^{S \times k}$ be a feature matrix. The effective dimension of Φ (vis-a-vis the standard basis (e_i)) is defined as the quantity*

$$d_{\text{eff}}(\Phi) := S \max_{i=1, \dots, S} \|P_\Phi e_i\|_2^2,$$

where P_Φ is the orthogonal projector onto the column space of Φ .

It is simple to check that the effective dimension is only a function of the column space of Φ and that $d_{\text{eff}}(\Phi)$ satisfies

$$\text{rank}(\Phi) \leq d_{\text{eff}}(\Phi) \leq S.$$

Our notion of effective dimension is derived from the *coherence* of Φ , defined as

$$\mu(\Phi) = \frac{d_{\text{eff}}}{\text{rank}(\Phi)}.$$

The notion of coherence is from Candès and Recht [2009], who demonstrate that coherence can be used to characterize the feasibility of low-rank matrix recovery. Informally, $\mu(\Phi)$ (and $d_{\text{eff}}(\Phi)$) measure the (lack of) sparsity of the column space of Φ . At one extreme, if $\Phi \in \mathbb{R}^{S \times 1}$ is the all-ones vector, then $d_{\text{eff}}(\Phi) = \text{rank}(\Phi)$,

saturating the lower bound. On the other hand, if $\Phi = e_i$ for some $i \in \{1, \dots, S\}$ then $d_{\text{eff}}(\Phi) = S$, saturating the upper bound. As we now show, the effective dimension of Φ can be used to bound the excess risk of least-squares regression applied to the state representation ϕ .

Theorem 6 (Excess risk). *Fix any $\delta \in (0, 1)$. Suppose that $n \geq 8d_{\text{eff}}(\Phi) \log(6k/\delta)$. With probability at least $1 - \delta$, the empirical risk minimizer $V_{\phi, \hat{w}}$ satisfies:*

$$\begin{aligned}
 \mathcal{E}(V_{\phi, \hat{w}}) &\leq \|P_{\Phi}^{\perp} V\|_{S,2}^2 + 384c \frac{d_{\text{eff}}(\Phi)}{n} \|P_{\Phi}^{\perp} V\|_{S,2}^2 + 48\sigma^2 \frac{2k + 3c}{n} \\
 &\quad + \frac{64}{3} \frac{d_{\text{eff}}(\Phi)}{n^2} \|P_{\Phi}^{\perp} V\|_{\infty}^2 c^2,
 \end{aligned}$$

where $c = \log(3/\delta)$ and $\|\cdot\|_{\infty}$ denotes the usual supremum norm.

Proof. The proof is given in Appendix 4.A, and follows arguments for the analysis of random design linear least-squares problems [Hsu et al., 2014] and matrix concentration inequalities [Tropp, 2015]. The result can also be obtained by instantiating Theorem 1 of Hsu et al. [2014] to our setting, at the cost of added complexity. \square

In Theorem 6, the term $\|P_{\Phi}^{\perp} V\|_{S,2}^2$ is the approximation error and reflects the error due to using a k -dimensional linear approximation. The remainder of the bound corresponds to the estimation error. The theorem demonstrates that the ability of a representation to generalize is quantified not only by the approximation error but also the effective dimension $d_{\text{eff}}(\Phi)$. Not only does $d_{\text{eff}}(\Phi)$ appear in the bound, but it also dictates a minimum number of samples needed to obtain a high probability bound: when $d_{\text{eff}}(\Phi)$ is small, the bound holds for fewer samples.

In the specific context of batch Monte Carlo policy evaluation, Theorem 6 holds as-is with $V = V^{\pi}$. Additionally, the noise variance σ^2 can be bounded as

$$\sigma^2 \leq \frac{V_{\max}^2}{4}.$$

The term $\|P_{\Phi} e_i\|_2^2$ that drives the effective dimension of Φ differs (for non orthogonal representations Φ) from the quantity $\max_i \|\phi(s_i)\|_2^2$ that appears in Rademacher complexity bounds for regression in the case of a family of linear

predictors [Mohri et al., 2018] (see also Maillard and Munos [2009]). Compared to such bounds, Theorem 6 is also sharper for all representations as it offers a $O(1/n)$ dependency rather than $O(1/\sqrt{n})$. In subsequent sections, we will provide empirical evidence illustrating how the effective dimension plays a critical role in determining the generalization capability of ϕ .

4.3.1 Illustrative Examples

To understand how the bound is instantiated in particular settings, consider first the scenario in which $\Phi = I_S$ is the tabular representation. This corresponds to using the feature vector $e_i \in \mathbb{R}^S$ for the i -th state. In this case, the approximation error is 0 and the estimation error reduces to the classic $\sigma^2 S/n$ rate for least-squares regression:

$$R(V_{\phi, \hat{w}}) - R(V) \lesssim \frac{\sigma^2(S + \log(1/\delta))}{n}.$$

With this choice of features, good generalization requires a number of samples n linear in S .

At the other extreme, it is possible to improve the sample complexity to avoid the dependency on S . In the ideal case, $d_{\text{eff}}(\Phi) = k$. In the next section we will demonstrate that, in environments with a particular transition structure, representations derived from the successor representation achieve this bound.

To make this argument more concrete, suppose that we have a family $(\phi_k)_{k=1}^S$ of representations (resp. matrices (Φ_k)) whose effective dimension satisfies $d_{\text{eff}}(\Phi_k) \approx k$. Furthermore, assume that the approximation error $\|P_{\Phi_k}^\perp V\|_{S,2}^2$ scales as $\psi(k)$, where $\psi(k)$ is a monotonically decreasing function of k . Fix $\varepsilon > 0$ and define $\bar{k} = \bar{k}(\varepsilon) := \min\{k : \psi(k) \leq \varepsilon\}$, and let \bar{w} be the weight vector found by least-squares regression applied with $\phi_{\bar{k}}$. Observe that as long as n satisfies:

$$n \gtrsim \max \left\{ \max \left\{ \frac{\sigma^2}{\varepsilon}, 1 \right\} \bar{k}(\varepsilon) \log \frac{\bar{k}(\varepsilon)}{\delta}, \sqrt{\bar{k}(\varepsilon) S} \log \frac{1}{\delta} \right\},$$

then we have $\mathcal{E}(V_{\phi_{\bar{k}}, \bar{w}}) \leq 4\varepsilon$. As a particular example, let $\psi(k) = \rho^k$ for some $\rho \in (0, 1)$. Then $\bar{k}(\varepsilon) \leq \lceil \frac{1}{1-\rho} \log \left(\frac{1}{\varepsilon} \right) \rceil$, in which case the sample complexity only depends sublinearly on S .

4.4 Generalization for the Successor Representation

An effective approach for constructing a family of representations is to take the k singular vectors of the successor representation (SR) whose singular values are the greatest. For a given policy π , let Ψ^π be the successor representation for π . We write

$$\Psi^\pi = F\Sigma B^\top,$$

where $F, B \in \mathbb{R}^{S \times S}$ are matrices whose columns are orthogonal and have unit norm. Additionally, $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_S)$ where σ_i are the singular values of Ψ sorted in decreasing order.

For a fixed integer k satisfying $1 \leq k \leq S$, let us partition F into two matrices, $F_k \in \mathbb{R}^{S \times k}$ and F_k^\perp , which respectively contains the top k and bottom $S - k$ columns of F . Correspondingly, we partition Σ into $\Sigma_k \in \mathbb{R}^{k \times k}$ and Σ_k^\perp and B into B_k and B_k^\perp . With this notation, we obtain the family of state representations (expressed as feature matrices) $\Phi_k = F_k$.

4.4.1 Approximation Error: $\|P_{\Phi}^\perp V^\pi\|_{S,2}^2$

Given a reward vector $r_\pi \in \mathbb{R}^S$, the value function $V^\pi \in \mathbb{R}^S$ is given by $V^\pi = \Psi^\pi r_\pi$. As demonstrated by Theorem 6, the first key quantity that appears in the generalization bound is the approximation error $\|P_{F_k}^\perp V^\pi\|_{S,2}^2$. With the successor representation, we can write:

$$\|P_{F_k}^\perp V^\pi\|_{S,2}^2 = \|P_{F_k}^\perp \Psi^\pi r_\pi\|_{S,2}^2 = \|F_k^\perp \Sigma_k^\perp (B_k^\perp)^\top r_\pi\|_{S,2}^2.$$

Following the argument from Petrik [2007] for the specific case of proto-value functions [Mahadevan and Maggioni, 2007], the worst-case unit-norm reward vector r_π in this case approximately corresponds to the $(k + 1)$ -th vector b_{k+1} . This is because

$$F_k^\perp \Sigma_k^\perp (B_k^\perp)^\top b_{k+1} = f_{k+1} \sigma_{k+1},$$

and the fact that $\sigma_{k+1} \geq \sigma_{k+i}$, for all $i \geq 1$. To make the bound comparable for different k and MDPs, let us fix R_{\max} and write

$$r_\pi = \frac{b_{k+1} R_{\max}}{\|b_{k+1}\|_\infty}. \quad (4.2)$$

In this case, since $\|f_{k+1}\|_2^2 = 1$, we have that

$$\|P_{F_k}^\perp V^\pi\|_{S,2}^2 \leq \frac{\sigma_{k+1}^2 R_{\max}^2}{S \|b_{k+1}\|_\infty^2} \leq \sigma_{k+1}^2 R_{\max}^2.$$

The dependence on $\|b_{k+1}\|_\infty$ relates to the operator norm of Ψ from L^2 to L^∞ , and illustrates how b_{k+1} is only approximately the worst-case reward vector.

A frequent scenario in reinforcement learning occurs when the reward is nonzero in a single state. Suppose that the reward vector r_π is $r_\pi = R_{\max} e_i$ for some $i \in \{1, \dots, S\}$. Then we have that:

$$\begin{aligned} \|P_{F_k}^\perp V^\pi\|_{S,2}^2 &= \frac{R_{\max}^2 \operatorname{tr}((\Sigma_k^\perp)^2) \|(B_k^\perp)^\top e_i\|_2^2}{S} \\ &\leq \frac{\sigma_{k+1}^2 R_{\max}^2 d_{\text{eff}}(B_k^\perp)}{S}. \end{aligned}$$

When the effective dimension of B_k^\perp is $O(S - k)$, the approximation error may be a factor $\frac{S-k}{S}$ smaller than the error for the worst-case reward vector (Equation (4.2)).

These arguments show that the generalization quality of a given family of representations can be partially quantified in terms of its spectrum $(\sigma_i)_{i=1}^S$. When the transition matrix is symmetric, we can bound the spectrum $(\sigma_i)_{i=1}^S$ in terms of the effective horizon implied by the discount factor. This is given by the following lemma.

Lemma 8. *Let $P \in \mathbb{R}^{S \times S}$ be a symmetric row stochastic matrix, and let $\gamma \in (0, 1)$. Let $\sigma(\cdot)$ denote the set of singular values of a matrix. We have that:*

$$\sigma((I - \gamma P)^{-1}) \subseteq \left[\frac{1}{1+\gamma}, \frac{1}{1-\gamma} \right].$$

Because the value function is generally of magnitude $V_{\max} = \frac{R_{\max}}{1-\gamma}$, an approximation error of order $\frac{1}{1+\gamma}$ is quite small, suggesting that the corresponding basis functions may be safely omitted from the representation.

Intuitively (and as supported by the analysis above), choosing a representation with a larger number of features k reduces the approximation error. However, as will see in the next section, a larger k necessarily increases the effective dimension, often in a manner that is superlinear in k .

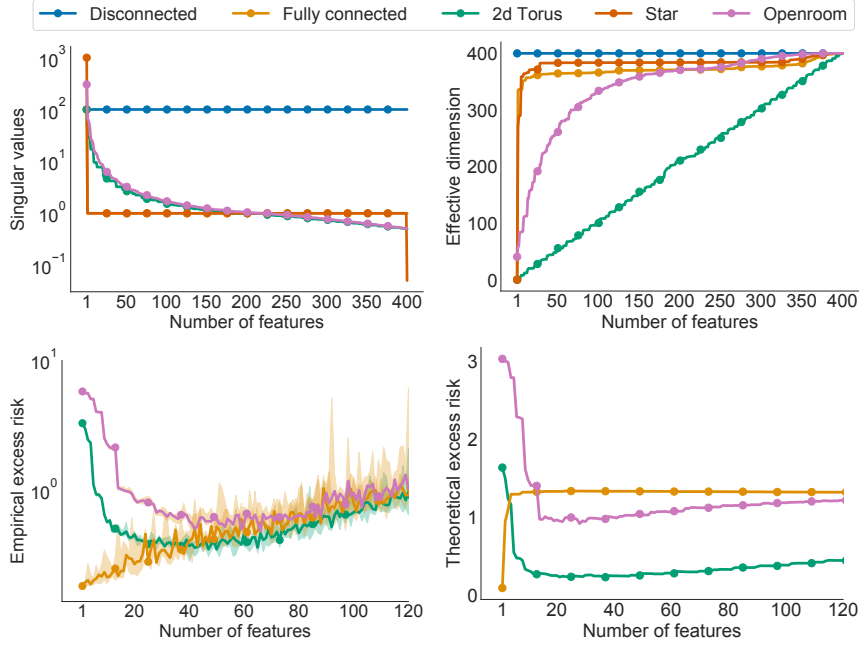


Figure 4.2: Singular values of the successor representation Ψ^π , in decreasing order and for different graphical structures. Note that the fully connected and star graphs’ spectra overlap (top left). Effective dimension of the representation $\Phi_k = F_k$ (top right). Median empirical excess risk over 10 runs, with 95% CIs as shaded regions, and theoretical excess risk, respectively, for the open room, torus, and fully connected graphs (bottom left and right).

4.4.2 Effect of Transition Structure

We next study characteristics of families of representations induced by the SVD of the successor representation for different environment transition structures. To this end, we consider different types of graphs over which we define a uniform random walk; the resulting representations are specifically proto-value functions [PVF, Mahadevan and Maggioni, 2007]. We consider the two key quantities identified above: the spectrum of the representation, which informs us on the profile of the approximation error $\|P_{F_k}^\top V^\pi\|_{S,2}^2$ for different F_k , and the effective dimension of F_k as a function of k .

We consider five graphical structures, each with $S = 400$ states (illustrations of these structures as well as results for additional structures are given in the appendix): a fully-connected graph, Baird’s star graph [Baird, 1995], a disconnected graph (on which each node self-transitions), a 20×20 grid, and a 20×20 torus. The torus has the same “shape” as the grid but allows transitions from one edge to its

opposite, while the fully-connected graph is similar to the star graph in that both mix quickly. These graph were chosen to illustrate the diversity in generalization profiles arising from different transition structures. In all cases, $\gamma = 0.99$.

Figure 4.2, top left illustrates three types of spectra. The fully-connected and star structures have a flat spectrum, both with an important first component but with a last component that is much smaller in the case of the star structure (see Appendix 4.B for a closed-form description of the spectrum of the star graph). By contrast, the grid and torus exhibit a decaying spectrum, suggesting that attaining a low approximation error may require many features. As expected, the disconnected graph produces a flat spectrum with values $\sigma_i = (1 - \gamma)^{-1}$.

Figure 4.2, top right shows the effective dimension as a function of the number of features k , and paints a relatively different picture. Here, both star and fully-connected graphs exhibit a high effective dimension, despite having relatively simple structure. This is because effective dimension reflects in some sense the degree to which a single sample might give misleading information about the value at other states. Because the first singular vectors capture most of the symmetry in these graphs, additional features must in some sense be misleading. On the other hand, the open room and torus, despite an almost-identical spectrum, exhibit notably different profiles: while the torus achieves the lower bound $d_{\text{eff}}(F_k) \approx k$, the grid results in generally poor features for k large.

To understand the consequences of these characteristic differences, we performed least-squares regression to estimate value functions in three of these structures (fully-connected, grid, and torus). In all cases, we sampled a reward function by assigning rewards to each state-action pair from a normal distribution (see Section 4.C). We then sampled $n = 300$ states with replacement and performed a Monte Carlo rollout to obtain the sample return $(y_i)_{i=1}^n$. We measured the excess risk of the linear approximation found by the least-squares procedure. For each graph structure, we repeated the experiment 10 times.

Figure 4.2, bottom depicts the outcome of this experiment. Experimentally, the PVF of the torus generalizes significantly better than the PVF of the grid

(left panel). This is reflected in a heuristic calculation of the theoretical bound (right panel), given more explicitly by the formula

$$\|P_{F_k}^\perp V^\pi\|_{S;2}^2 + \frac{d_{\text{eff}}(F_k)}{n} + \frac{d_{\text{eff}}(F)}{n^2} \|P_{F_k}^\perp V^\pi\|_\infty^2.$$

The number of features k minimizing the empirical and theoretical excess risk differ, but follow the same qualitative pattern: for small k , the open room PVF generalizes poorly, while the minimum is achieved in the fully-connected graph by $k = 1$, highlighting again its high degree of symmetry.

4.4.3 Analysis of the One-dimensional Torus

As evidenced by the experiments of the previous section, the proto-value functions of the two-dimensional torus have particularly appealing generalization characteristics. Analytically, similarly good generalization can be demonstrated on the one-dimensional torus, as we now show.

The one-dimensional torus consists in S states arranged on a chain, such that s_i connects to $s_{i-1}, s_{i+1} \bmod S$. As such, the random walk on this torus induces a transition function P_π described by a circulant matrix. Since P_π is symmetric, we may write²

$$(I - \gamma P_\pi)^{-1} = U_S \Sigma U_S^*.$$

Following Gray et al. [2006], the k -th singular value of $(I - \gamma P_\pi)^{-1}$ is given by

$$\sigma_k = \frac{1}{1 - \gamma \cos(\frac{2\pi}{S} \lceil \frac{k-1}{2} \rceil)}$$

for $k = 1, \dots, S$.³ and we have that $U_S = \frac{1}{\sqrt{S}} F_S^*$, with $(F_S)_{j,k} = \exp(-2\pi i j k / S)$ the discrete Fourier transform matrix in dimension S . From this we deduce that each entry of U_S has modulus $1/\sqrt{S}$, and therefore any orthogonal matrix formed from any k distinct columns of U_S will have coherence 1 and effective dimension k . This shows that the proto-value functions of the one-dimensional torus give in some sense an ideal state representation.

²We ignore the issue of real diagonalizable versus complex diagonalizable.

³The spectrum of the torus is briefly mentioned in Blier et al. [2021].

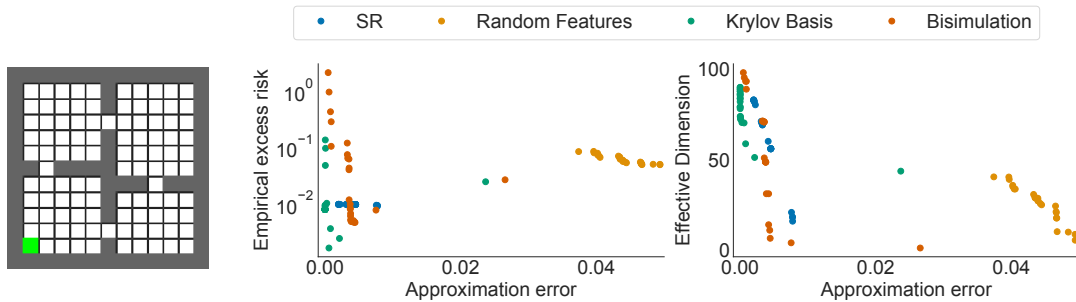


Figure 4.3: The Four Rooms domain (left). Median empirical excess risk (middle) and effective dimension (right) as a function of approximation error for the top k left singular vectors of the SR, random features, the Krylov basis and the bisimulation metric matrix in the Four rooms domain.

4.5 Experiments

4.5.1 Comparing State Representations

We now compare the Successor Representation to other theoretically-motivated representations: the bisimulation metric matrix (Ferns et al., 2004), the Krylov basis (Petrik, 2007) and some random features, in terms of effective dimension and excess risk, in the setting of Section 4.2. Figure 4.3 shows some of these results on the Four Rooms domain [Sutton et al., 1999, Solway et al., 2014]. These give further weight to the idea that effective dimension plays an important role in determining the usefulness of a representation, as for a given approximation error better effective dimension corresponds to better excess risk.

The SR of the Four Rooms domain is fairly well-studied and have been shown to give rise to effective representations [Machado et al., 2017, Bellemare et al., 2019]. It generalizes well but has worse approximation error compared to the Krylov basis or the Bisimulation metric which take into account the reward. For small approximation errors, the krylov basis has smaller effective dimension and is performing best. Finally, random features which are agnostic to the structure of the MDP have very high approximation error making them unappealing.

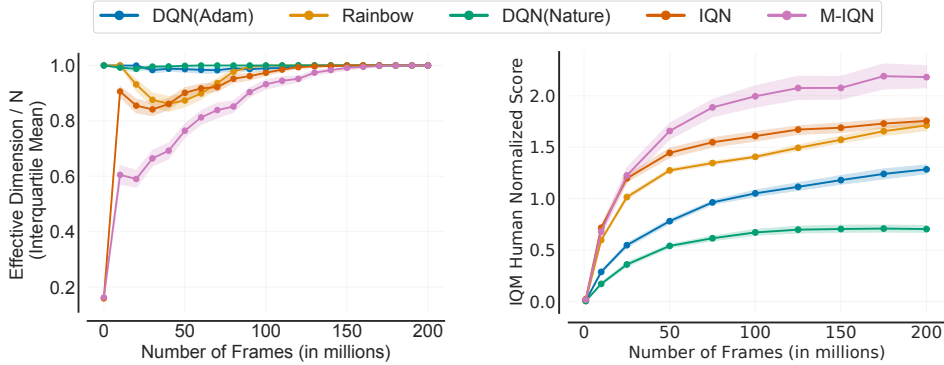


Figure 4.4: Interquartile mean (IQM) [Agarwal et al., 2021b] for the effective dimension, normalized by the batch size used $N = 2^{15}$ (left). Interquartile mean (IQM) for human-normalized scores over the course of training across 60 Atari games (right). IQM measures the mean on the middle 50% of the data points combined across all runs and games. These statistics are over 5 independent runs and shading gives 95% stratified bootstrap confidence intervals based on Rliable [Agarwal et al., 2021b].

4.5.2 Deep Reinforcement Learning

We conclude with an empirical evaluation demonstrating the usefulness of our results in characterizing generalization in a larger setting. Specifically, we measure the effective dimensions of a representation ϕ implied by a deep neural network. We consider the hidden layer of 512 rectified linear units learnt by five deep RL agents, namely DQN [Mnih et al., 2015], DQN with Adam optimizer, Rainbow [Hessel et al., 2018], IQN [Dabney et al., 2018a], and Munchausen-IQN (M-IQN) [Veillard et al., 2020]. We are interested in how the notion of effective dimension explains the relative performance of these deep RL agents aggregated across 60 Atari 2600 games [Bellemare et al., 2013] and at different points in training until 200M environment frames [Castro et al., 2018].

We compare estimates of the effective dimension of these representations throughout training and reported results in Figure 4.4 (Left) (see per game comparison in Appendix 4.C.2). When computing such estimates, we use a large batch size ($=2^{15}$), sampled uniformly from the offline Atari-replay datasets [Agarwal et al., 2020], as a proxy for the ambient dimension S used in the definition of the effective dimension.

We observe that higher performance on a game typically correlates with lower effective dimension. The relative ordering of effective dimension (Figure 4.4, left)

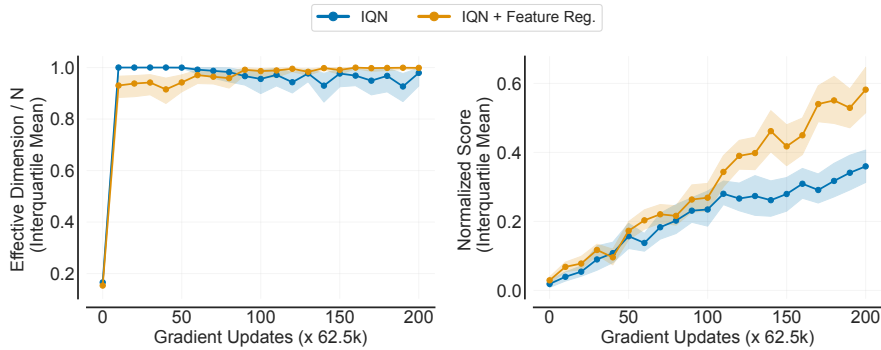


Figure 4.5: Effective dimension, normalized by the batch size $N = 2^{15}$ and performance of IQN and IQN with feature regularization L_ϕ on 17 Atari games in the offline RL setting.

matches the performance ranking of different agents (Figure 4.4, right). Furthermore, we can notice a rise in the effective dimension from iteration 50 which suggests an overfitting of the representation to the current value function, in line with the evidence of late-training overfitting found by Dabney et al. [2021].

To further corroborate that low effective dimension corresponds to better generalization, we investigate whether optimizing an auxiliary loss \mathcal{L}_ϕ , motivated by the idea of reducing the effective dimension of the learned representation, improves performance. To do so, we use $\mathcal{L}_\phi = \log \sum_i \exp(\|\phi(s_i)\|_2^2)$ for states s_i in a randomly sampled mini-batch of size 32. To avoid confounding effects from exploration, we study the offline RL setting [Levine et al., 2020]. Specifically, we use the 5% Atari-replay dataset [Agarwal et al., 2020] on 17 games and evaluate IQN, one of the top performing agents on the offline Atari dataset [Gulcehre et al., 2020]. As shown in Figure 4.5, right, combining IQN with the loss \mathcal{L}_ϕ results in significantly higher average returns compared to IQN on all 17 games. We also compare estimates of the effective dimension of the representations induced by these two agents in Figure 4.5, left, and find the auxiliary loss \mathcal{L}_ϕ results in lower effective dimension during the first 80 iterations. Surprisingly, we also notice that IQN with feature regularization prevents the substantial loss in rank of the feature matrix observed previously by Kumar et al. [2021, 2022] (see Figure 4.13 and Figure 4.12), making it hard to disentangle between approximation and estimation error effects. Further study of this phenomenon would be an interesting direction for future work.

4.6 Conclusion

In this paper we provided a theoretical characterisation of how a given representation affects generalization in reinforcement learning. While we focused here on the batch Monte Carlo setting for simplicity, a similar but more involved analysis can in theory also be performed to analyze algorithms such as LSTD.

Providing fresh evidence regarding the benefits of successor representations in shaping an agent’s representation, both our analysis and experiments on synthetic environments demonstrate that indeed, the left-singular vectors of SRs generally provide good generalization. While natural given the successor representation’s close relationship with the value function, one surprising result is that the effective dimension of such a representation is relatively sensitive to the particular transition structure, as illustrated by the differences between the torus and open room representations. In addition, the effective dimension of this representation does not immediately correlate with mixing time, as one might have expected. These findings suggest that it should be possible to devise algorithms inspired by the same principles, but that work well across a variety of transition structures, for example by leveraging contrastive graph representations [Madjiheurem and Toni, 2019].

Our analysis of Atari 2600-playing agents gives further evidence of the important role played by the representation in deep reinforcement learning. While not a surprise in itself, we find a strong correlation between effective dimension and performance, this suggests that generalization is key to explaining many performance improvements. In particular, it is by now well-understood that auxiliary tasks [Jaderberg et al., 2017, Bellemare et al., 2017] shape the learned representation of the agent, and under ideal conditions cause it to match the SVD of an auxiliary task matrix [Bellemare et al., 2019, Lyle et al., 2021]. Controlling the bound of Theorem 6 by means of such tasks or deep learning mechanisms such as hindsight experience replay [Andrychowicz et al., 2017] may provide further performance improvements. Our results also suggest that it may be possible to derive theoretical

guarantees regarding transfer between policies or MDPs [Taylor and Stone, 2009], in particular with a learned representation [Agarwal et al., 2021a].

Acknowledgements

The authors would like to thank Matthieu Geist, Mark Rowland, Pablo Samuel Castro, Ahmed Touati, Marlos Machado, Dale Schuurmans, Robert Dadashi, Tomas Vaskevicius, Olivier Pietquin, Martha White, Hanie Sedghi, Damien Vincent, Dominic Richards, Nino Vieillard, Leonard Hussenot, Amartya Sanyal, Sephora Madjiheurem, Laura Toni and the anonymous reviewers for useful discussions and feedback on this paper.

We also thank the Python community [Van Rossum and Drake Jr, 1995, Oliphant, 2007] for developing tools that enabled this work, including NumPy [Oliphant, 2006, Walt et al., 2011, Harris et al., 2020], SciPy [Jones et al., 2001], Matplotlib [Hunter, 2007] and JAX [Bradbury et al., 2018].

4.A Proofs for Section 4.3

This section is dedicated to proving the main theorem on the paper, Theorem 6. Before that, we introduce and prove a more general result from which Theorem 6 can be deduced as a corollary.

Let s_1, \dots, s_n denote iid draws from an arbitrary distribution $\nu \in \mathcal{P}(\mathcal{S})$ and $(e_i)_{i=1}^S \subset \mathbb{R}^S$ the standard basis.

Assumption 1. *We assume that $\nu(s) > 0$ for all state $s \in \{1, \dots, S\}$.*

Let $N := \mathbb{E}_{i \sim \nu}[e_i e_i^\top]$, and let $\|x\|_{\nu,2} := \|N^{1/2}x\|_2$ for $x \in \mathbb{R}^S$. Put $\underline{\nu} := \min_{i=1, \dots, S} \nu_i > 0$. Let $w^* := (\Phi^\top N \Phi)^{-1} \Phi^\top N V$, and also define $\Xi := \Phi^\top N \Phi$. Ξ is the steady-state feature covariance matrix. w^* represents the best k -dimensional model. Since we assume that $\underline{\nu} > 0$, we have that Ξ is positive definite.

The excess risk $\mathcal{E}(V_{\phi,w})$ of a hypothesis $V_{\phi,w} : \mathcal{S} \rightarrow \mathbb{R}$ is defined as:

$$\mathcal{E}(V_{\phi,w}) := \mathbb{E}_{s_i \sim \nu} (V_{\phi,w}(s_i) - V(s_i))^2.$$

For any $\hat{w} \in \mathbb{R}^k$, we have the decomposition:

$$\mathcal{E}(V_{\phi, \hat{w}}) = \|\Phi \hat{w} - V\|_{\nu, 2}^2 = \|\Phi(\hat{w} - w^*)\|_{\nu, 2}^2 + \|\Phi w^* - V\|_{\nu, 2}^2.$$

Note we have the identity:

$$\|\Phi w^* - V\|_{\nu, 2}^2 = \|P_{N^{1/2}\Phi}^\perp N^{1/2}V\|_2^2.$$

Theorem 7. Fix any $\delta \in (0, 1)$. Suppose that $n \geq 8d_{\text{eff}}(\Phi) \log(6k/\delta)$. Under Assumption 1, with probability at least $1 - \delta$, the empirical risk minimizer $V_{\phi, \hat{w}}$ satisfies:

$$\begin{aligned} \mathcal{E}(V_{\phi, \hat{w}}) &\leq \|P_{N^{1/2}\Phi}^\perp N^{1/2}V\|_2^2 + 384 \frac{d_{\text{eff}}(\Phi)}{\underline{\nu}nS} \|P_{N^{1/2}\Phi}^\perp N^{1/2}V\|_2^2 \log(3/\delta) \\ &\quad + 48 \frac{\sigma^2}{n} [2k + 3 \log(3/\delta)] + \frac{64}{3} \frac{d_{\text{eff}}(\Phi)}{\underline{\nu}n^2S} \|N^{-1/2}P_{N^{1/2}\Phi}^\perp N^{1/2}V\|_\infty^2 \log^2(3/\delta), \end{aligned}$$

where $\|\cdot\|_\infty$ denotes the usual supremum norm.

Proof. The empirical risk minimizer $\hat{w} \in \mathbb{R}^k$ is defined as the random vector $\hat{w} = (E_n \Phi)^\dagger Y$. Next, we write:

$$N^{1/2}\Phi(\hat{w} - w^*) = N^{1/2}\Phi(E_n \Phi)^\dagger (E_n V + \eta) - N^{1/2}\Phi w^*$$

Therefore, assuming $E_n \Phi$ has full column rank (which will be the case by Lemma 9),

$$\begin{aligned} &N^{1/2}\Phi(E_n \Phi)^\dagger E_n V - N^{1/2}\Phi w^* \\ &= N^{1/2}\Phi(E_n \Phi)^\dagger E_n V - P_{N^{1/2}\Phi} N^{1/2}V \\ &= N^{1/2}\Phi(\Phi^\top E_n^\top E_n \Phi)^{-1} \Phi^\top E_n^\top E_n V - P_{N^{1/2}\Phi} N^{1/2}V \\ &= N^{1/2}\Phi(\Phi^\top E_n^\top E_n \Phi)^{-1} \Phi^\top E_n^\top E_n N^{-1/2} (P_{N^{1/2}\Phi} + P_{N^{1/2}\Phi}^\perp) N^{1/2}V - P_{N^{1/2}\Phi} N^{1/2}V \\ &= N^{1/2}\Phi(\Phi^\top E_n^\top E_n \Phi)^{-1} \Phi^\top E_n^\top E_n N^{-1/2} P_{N^{1/2}\Phi}^\perp N^{1/2}V \\ &\quad + N^{1/2}\Phi(\Phi^\top E_n^\top E_n \Phi)^{-1} \Phi^\top E_n^\top E_n N^{-1/2} P_{N^{1/2}\Phi} N^{1/2}V - P_{N^{1/2}\Phi} N^{1/2}V \\ &= N^{1/2}\Phi(\Phi^\top E_n^\top E_n \Phi)^{-1} \Phi^\top E_n^\top E_n N^{-1/2} P_{N^{1/2}\Phi}^\perp N^{1/2}V \\ &= N^{1/2}\Phi \Xi^{-1/2} (\Xi^{-1/2} \Phi^\top E_n^\top E_n \Phi \Xi^{-1/2})^{-1} \Xi^{-1/2} \Phi^\top E_n^\top E_n N^{-1/2} P_{N^{1/2}\Phi}^\perp N^{1/2}V. \end{aligned}$$

Similarly,

$$\begin{aligned} N^{1/2}\Phi(E_n\Phi)^\dagger\eta &= N^{1/2}\Phi(\Phi^\top E_n^\top E_n\Phi)^{-1}\Phi^\top E_n^\top\eta \\ &= N^{1/2}\Phi\Xi^{-1/2}(\Xi^{-1/2}\Phi^\top E_n^\top E_n\Phi\Xi^{-1/2})^{-1}\Xi^{-1/2}\Phi^\top E_n^\top\eta. \end{aligned}$$

We first claim that $\|N^{1/2}\Phi\Xi^{-1/2}\|_{\text{op}} \leq 1$. To see this, observe that:

$$\|N^{1/2}\Phi\Xi^{-1/2}\|_{\text{op}}^2 = \lambda_{\max}(N^{1/2}\Phi(\Phi^\top N\Phi)^{-1}\Phi^\top N^{1/2}) = \lambda_{\max}(P_{N^{1/2}\Phi}) \leq 1.$$

Hence:

$$\|N^{1/2}\Phi(E_n\Phi)^\dagger E_n V - N^{1/2}\Phi w_*\|_2 \leq \frac{\|\Xi^{-1/2}\Phi^\top E_n^\top E_n N^{-1/2} P_{N^{1/2}\Phi}^\perp N^{1/2} V\|_2}{\lambda_{\min}(\Xi^{-1/2}\Phi^\top E_n^\top E_n\Phi\Xi^{-1/2})},$$

and similarly

$$\|N^{1/2}\Phi(E_n\Phi)^\dagger\eta\|_2 \leq \frac{\|\Xi^{-1/2}\Phi^\top E_n^\top\eta\|_2}{\lambda_{\min}(\Xi^{-1/2}\Phi^\top E_n^\top E_n\Phi\Xi^{-1/2})}.$$

Therefore,

$$\|N^{1/2}\Phi(\hat{w} - w^*)\|_2 \leq \frac{[\|\Xi^{-1/2}\Phi^\top E_n^\top E_n N^{-1/2} P_{N^{1/2}\Phi}^\perp N^{1/2} V\|_2 + \|\Xi^{-1/2}\Phi^\top E_n^\top\eta\|_2]}{\lambda_{\min}(\Xi^{-1/2}\Phi^\top E_n^\top E_n\Phi\Xi^{-1/2})}$$

By Lemma 9, as long as $n \geq \frac{8d_{\text{eff}}(\Phi)}{\underline{\nu}S} \log(6k/\delta)$, then with probability at least $1 - \delta/3$,

$$\frac{n}{2}I_k \preceq \Xi^{-1/2}\Phi^\top E_n^\top E_n\Phi\Xi^{-1/2} \preceq 4nI_k.$$

Furthermore, by Lemma 10, with probability at least $1 - \delta/3$,

$$\begin{aligned} \|\Xi^{-1/2}\Phi^\top E_n^\top E_n N^{-1/2} P_{N^{1/2}\Phi}^\perp N^{1/2} V\|_2 &\leq 2\sqrt{\frac{8nd_{\text{eff}}(\Phi)}{\underline{\nu}S}} \|P_{N^{1/2}\Phi}^\perp N^{1/2} V\|_2^2 \log\left(\frac{3}{\delta}\right) \\ &\quad + \frac{4}{3}\sqrt{\frac{d_{\text{eff}}(\Phi)}{\underline{\nu}S}} \|N^{-1/2} P_{N^{1/2}\Phi}^\perp N^{1/2} V\|_\infty \log\left(\frac{3}{\delta}\right) \end{aligned}$$

Finally, by Lemma 11, with probability at least $1 - \delta/3$,

$$\mathbf{1} \left\{ \Xi^{-1/2}\Phi^\top E_n^\top E_n\Phi\Xi^{-1/2} \preceq 4nI_k \right\} \cdot \|\Xi^{-1/2}\Phi^\top E_n^\top\eta\|_2 \leq \sqrt{\sigma^2 n [8k + 12 \log(3/\delta)]}.$$

Therefore, by a union bound, with probability at least $1 - \delta$,

$$\begin{aligned}
\|N^{1/2}\Phi(\hat{w} - w^*)\|_2 &\leq \frac{2}{n} \left[2\sqrt{\frac{8nd_{\text{eff}}(\Phi)}{\underline{\nu}S}} \|P_{N^{1/2}\Phi}^\perp N^{1/2}V\|_2^2 \log\left(\frac{3}{\delta}\right) \right] \\
&\quad + \frac{2}{n} \left[\frac{4}{3}\sqrt{\frac{d_{\text{eff}}(\Phi)}{\underline{\nu}S}} \|N^{-1/2}P_{N^{1/2}\Phi}^\perp N^{1/2}V\|_\infty \log\left(\frac{3}{\delta}\right) \right] \\
&\quad + \frac{2}{n} \left[\sqrt{\sigma^2 n [8k + 12 \log(3/\delta)]} \right] \\
&= 4\sqrt{8}\sqrt{\frac{d_{\text{eff}}(\Phi)}{\underline{\nu}nS}} \log(3/\delta) \|P_{N^{1/2}\Phi}^\perp N^{1/2}V\|_2 \\
&\quad + 4\sqrt{\frac{\sigma^2}{n} [2k + 3 \log(3/\delta)]} \\
&\quad + \frac{8}{3} \sqrt{\frac{d_{\text{eff}}(\Phi)}{\underline{\nu}S}} \|N^{-1/2}P_{N^{1/2}\Phi}^\perp N^{1/2}V\|_\infty \log\left(\frac{3}{\delta}\right).
\end{aligned}$$

Now, from the inequality $(a+b+c)^2 \leq 3(a^2 + b^2 + c^2)$ for any $a, b, c \in \mathbb{R}$, it follows that

$$\begin{aligned}
\mathcal{E}(V_{\phi, \hat{w}}) &= \|P_{N^{1/2}\Phi}^\perp N^{1/2}V\|_2^2 + 384 \frac{d_{\text{eff}}(\Phi)}{\underline{\nu}nS} \|P_{N^{1/2}\Phi}^\perp N^{1/2}V\|_2^2 \log(3/\delta) \\
&\quad + 48 \frac{\sigma^2}{n} [2k + 3 \log(3/\delta)] + \frac{64}{3} \frac{d_{\text{eff}}(\Phi)}{\underline{\nu}n^2S} \|N^{-1/2}P_{N^{1/2}\Phi}^\perp N^{1/2}V\|_\infty^2 \log^2(3/\delta).
\end{aligned}$$

□

Lemma 9. Let $\Phi \in \mathbb{R}^{S \times k}$. Let ν denote a distribution over $\{1, \dots, S\}$ satisfying Assumption 1 and $(e_i)_{i=1}^S \subset \mathbb{R}^S$ the standard basis. Let s_1, \dots, s_n denote iid draws from ν . Define $Y_n \in \mathbb{R}^{k \times k}$ as:

$$Y_n = \sum_{i=1}^n \Xi^{-1/2} \Phi^\top e_{s_i} e_{s_i}^\top \Phi \Xi^{-1/2}.$$

Fix any $\delta \in (0, 1)$. As long as $n \geq \frac{8d_{\text{eff}}(\Phi)}{\underline{\nu}S} \log(2k/\delta)$, with probability at least $1 - \delta$,

$$\frac{n}{2} I_k \preceq Y_n \preceq 4n I_k.$$

where for two symmetric matrices, $A \preceq B$ means that the matrix $B - A$ is positive semi-definite.

Proof. This is an application of the Matrix Chernoff inequality. First, we see that $\mathbb{E}[Y_n] = nI_k$. Next, we have:

$$\begin{aligned} \max_{i=1,\dots,S} \lambda_{\max}(\Xi^{-1/2} \Phi^\top e_i e_i^\top \Phi \Xi^{-1/2}) &= \max_{i=1,\dots,S} \|\Xi^{-1/2} \Phi^\top e_i\|_2^2 \\ &= \max_{i=1,\dots,S} \|(\Phi^\top N \Phi)^{-1/2} \Phi^\top e_i\|_2^2 \\ &\leq \frac{1}{\underline{\nu}} \max_{i=1,\dots,S} \|P_\Phi e_i\|_2^2 \\ &\leq \frac{d_{\text{eff}}(\Phi)}{\underline{\nu}S}. \end{aligned}$$

We now make two applications of the Matrix Chernoff inequality (see Theorem 5.1.1 in Tropp [2015]). Denoting e as Euler's number, for the upper tail, we have that for any $t \geq e$,

$$\mathbb{P}(\lambda_{\max}(Y_n) \geq tn) \leq k(e/t)^{tn\underline{\nu}S/d_{\text{eff}}(\Phi)}.$$

Setting $t = 4$, we conclude that as long as $n \geq \frac{1}{4 \log(4/e)} \frac{d_{\text{eff}}(\Phi)}{\underline{\nu}S} \log(2k/\delta)$, then we have that with probability at least $1 - \delta/2$, $\lambda_{\max}(Y_n) \leq 4n$. For the lower tail, we have that for any $t \in (0, 1)$,

$$\mathbb{P}(\lambda_{\min}(Y_n) \leq tn) \leq k \exp\left(- (1-t)^2 \frac{n}{2} \frac{\underline{\nu}S}{d_{\text{eff}}(\Phi)}\right).$$

Setting $t = 0.5$, we see that as long as $n \geq 8 \frac{d_{\text{eff}}(\Phi)}{\underline{\nu}S} \log(2k/\delta)$, then $\lambda_{\min}(Y_n) \geq n/2$ with probability at least $1 - \delta/2$. Taking a union bound yields the claim. \square

Lemma 10. Put $z_n := \Xi^{-1/2} \Phi^\top E_n^\top E_n N^{-1/2} P_{N^{1/2}\Phi}^\perp N^{1/2} V$. Fix any $\delta \in (0, e^{-1/8})$.

With probability at least $1 - \delta$,

$$\begin{aligned} \|z_n\|_2 &\leq 2 \sqrt{\frac{8nd_{\text{eff}}(\Phi)}{\underline{\nu}S}} \|P_{N^{1/2}\Phi}^\perp N^{1/2} V\|_2 \sqrt{\log(1/\delta)} \\ &\quad + \frac{4}{3} \sqrt{\frac{d_{\text{eff}}(\Phi)}{\underline{\nu}S}} \|N^{-1/2} P_{N^{1/2}\Phi}^\perp N^{1/2} V\|_\infty \log(1/\delta). \end{aligned}$$

Proof. Define $q_i := \Xi^{-1/2} \Phi^\top e_{s_i} e_{s_i}^\top N^{-1/2} P_{N^{1/2}\Phi}^\perp N^{1/2} V$. We have that $\mathbb{E}[q_i] = 0$. Next,

$$\begin{aligned} \mathbb{E}[\|q_i\|_2^2] &= \mathbb{E}[\|\Xi^{-1/2} \Phi^\top e_{s_i}\|_2^2 \langle e_{s_i}, N^{-1/2} P_{N^{1/2}\Phi}^\perp N^{1/2} V \rangle^2] \\ &\leq \frac{d_{\text{eff}}(\Phi)}{\underline{\nu}S} \mathbb{E}[\langle e_{s_i}, N^{-1/2} P_{N^{1/2}\Phi}^\perp N^{1/2} V \rangle^2] \\ &= \frac{d_{\text{eff}}(\Phi)}{\underline{\nu}S} \|P_{N^{1/2}\Phi}^\perp N^{1/2} V\|_2^2. \end{aligned}$$

Finally, we have the following almost sure bound:

$$\|q_i\|_2 \leq \sqrt{\frac{d_{\text{eff}}(\Phi)}{\underline{\nu}S}} \|N^{-1/2} P_{N^{1/2}\Phi}^\perp N^{1/2} V\|_\infty.$$

Put $z_n := \sum_{i=1}^n q_i$. By the vector Bernstein inequality, for all $t > 0$,

$$\mathbb{P}(\|z_n\|_2 > Z_t) \leq e^{-t}.$$

where $Z_t = \sqrt{\frac{nd_{\text{eff}}(\Phi)}{\underline{\nu}S} \|P_{N^{1/2}\Phi}^\perp N^{1/2} V\|_2^2 (1 + \sqrt{8t})} + \frac{4}{3} \sqrt{\frac{d_{\text{eff}}(\Phi)}{\underline{\nu}S}} \|N^{-1/2} P_{N^{1/2}\Phi}^\perp N^{1/2} V\|_\infty t$.

The claim now follows by setting $t = \log(1/\delta)$. \square

Lemma 11. *Let \mathcal{G} be the event:*

$$\mathcal{G} := \left\{ \Xi^{-1/2} \Phi^\top E_n^\top E_n \Phi \Xi^{-1/2} \preceq 4nI_k \right\}$$

With probability at least $1 - \delta$, we have:

$$\mathbf{1}\{\mathcal{G}\} \cdot \|\Xi^{-1/2} \Phi^\top E_n^\top \eta\|_2^2 \leq \sigma^2 n [8k + 12 \log(1/\delta)].$$

Proof. Put $M := \mathbf{1}\{\mathcal{G}\} \cdot E_n \Phi \Xi^{-1} \Phi^\top E_n^\top$. Because η is assumed to be independent of E_n , we can condition on E_n and apply the Hanson-Wright inequality [Hsu et al., 2012] to conclude that for any $t > 0$,

$$\mathbb{P}(\eta^\top M \eta > \sigma^2 (\text{tr}(M) + 2\sqrt{\text{tr}(M^2)t} + 2\|M\|_{\text{op}}t) \mid E_n) \leq e^{-t}.$$

We now compute upper bounds on $\text{tr}(M)$, $\text{tr}(M^2)$, and $\|M\|_{\text{op}}$. First, we have:

$$\text{tr}(M) = \mathbf{1}\{\mathcal{G}\} \text{tr}(E_n \Phi \Xi^{-1} \Phi^\top E_n^\top) = \mathbf{1}\{\mathcal{G}\} \text{tr}(\Xi^{-1/2} \Phi^\top E_n^\top E_n \Phi \Xi^{-1/2}) \leq 4nk.$$

Next,

$$\begin{aligned} \text{tr}(M^2) &= \mathbf{1}\{\mathcal{G}\} \text{tr}(E_n \Phi \Xi^{-1} \Phi^\top E_n^\top E_n \Phi \Xi^{-1} \Phi^\top E_n^\top) \\ &= \mathbf{1}\{\mathcal{G}\} \text{tr}(\Xi^{-1/2} \Phi^\top E_n^\top E_n \Phi \Xi^{-1/2} \cdot \Xi^{-1/2} \Phi^\top E_n^\top E_n \Phi \Xi^{-1/2}) \\ &\stackrel{(a)}{\leq} \mathbf{1}\{\mathcal{G}\} \text{tr}(\Xi^{-1/2} \Phi^\top E_n^\top E_n \Xi^{-1/2} \Phi) \|\Xi^{-1/2} \Phi^\top E_n^\top E_n \Phi \Xi^{-1/2}\|_{\text{op}} \\ &\leq 4nk \cdot 4n = 16n^2k. \end{aligned}$$

Above, (a) follows from Hölder's inequality. Finally,

$$\|M\|_{\text{op}} = \mathbf{1}\{\mathcal{G}\} \|E_n \Phi \Xi^{-1} \Phi^\top E_n^\top\|_{\text{op}} = \mathbf{1}\{\mathcal{G}\} \|\Xi^{-1/2} \Phi^\top E_n^\top E_n \Phi \Xi^{-1/2}\|_{\text{op}} \leq 4n.$$

We now plug these bounds in along with the choice of $t = \log(1/\delta)$, which tells us that conditioned on E_n , with probability at least $1 - \delta$,

$$\begin{aligned} \eta^\top M \eta &\leq \sigma^2 \left[4nk + 8n\sqrt{k \log(1/\delta)} + 8n \log(1/\delta) \right] \\ &\leq \sigma^2 [8nk + 12n \log(1/\delta)] \\ &= \sigma^2 n [8k + 12 \log(1/\delta)]. \end{aligned}$$

We now remove the conditioning on E_n . Let $\bar{t} := \sigma^2 n [8k + 12 \log(1/\delta)]$. By the tower property,

$$\begin{aligned} \mathbb{P}(\eta^\top M \eta \geq \bar{t}) &= \mathbb{E}[\mathbf{1}\{\eta^\top M \eta \geq \bar{t}\}] = \mathbb{E}[\mathbb{E}[\mathbf{1}\{\eta^\top M \eta \geq \bar{t}\} \mid E_n]] \\ &= \mathbb{E}[\mathbb{P}(\eta^\top M \eta \geq \bar{t} \mid E_n)] \leq \mathbb{E}[\delta] = \delta. \end{aligned}$$

□

Theorem 6 is a corollary of Theorem 7 in the case where the distribution ν is uniform.

Theorem 6 (Excess risk). *Fix any $\delta \in (0, 1)$. Suppose that $n \geq 8d_{\text{eff}}(\Phi) \log(6k/\delta)$. With probability at least $1 - \delta$, the empirical risk minimizer $V_{\phi, \hat{w}}$ satisfies:*

$$\begin{aligned} \mathcal{E}(V_{\phi, \hat{w}}) &\leq \|P_\Phi^\perp V\|_{S,2}^2 + 384c \frac{d_{\text{eff}}(\Phi)}{n} \|P_\Phi^\perp V\|_{S,2}^2 + 48\sigma^2 \frac{2k + 3c}{n} \\ &\quad + \frac{64}{3} \frac{d_{\text{eff}}(\Phi)}{n^2} \|P_\Phi^\perp V\|_\infty^2 c^2, \end{aligned}$$

where $c = \log(3/\delta)$ and $\|\cdot\|_\infty$ denotes the usual supremum norm.

Proof. ν being uniform, we have $\underline{\nu} = S$. The result follows by plugging $\underline{\nu}$ in Theorem 7. □

4.B Proofs for Section 4.4

Lemma 8. *Let $P \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ be a symmetric row stochastic matrix, and let $\gamma \in (0, 1)$. Let $\sigma(\cdot)$ denote the set of singular values of a matrix. We have that:*

$$\sigma((I - \gamma P)^{-1}) \subseteq \left[\frac{1}{1+\gamma}, \frac{1}{1-\gamma} \right].$$

Proof. Let $\lambda(\cdot)$ denote the eigenvalues of a matrix. Because P is symmetric, we have that:

$$\sigma((I - \gamma P)^{-1}) = \left\{ \frac{1}{1 - \gamma\lambda} : \lambda \in \lambda(P) \right\}.$$

Because P is a row stochastic matrix, we have that the spectral radius of P satisfies $\rho(P) = 1$, and therefore $\lambda(P) \subseteq [-1, 1]$. Hence:

$$\frac{1}{1 - \gamma\lambda} \in [1/(1 + \gamma), 1/(1 - \gamma)].$$

□

Eigenstructure of the Star Graph (Subsection 4.4.2)

A random walk on the Star graph induces a rank-two transition matrix $P_\pi \in \mathbb{R}^S$. We may write $P_\pi = v_1 e_S^\top + \frac{e_S v^\top}{S-1}$ where v is an all-ones vector except on its last coordinate where it takes value 0 and e_S a one-hot vector taking value 1 on its last coordinate. It is easy to prove by induction that

- for any $k \geq 1$, $P_\pi^{2k} = \frac{v v^\top}{S-1} + e_S e_S^\top$
- for any $k \geq 0$, $P_\pi^{2k+1} = P_\pi$

From this, it follows that

$$\begin{aligned} (I - \gamma P_\pi)^{-1} &= I + \sum_{t=1}^{\infty} (\gamma P_\pi)^t \\ &= I + \sum_{2k \geq 2} \gamma^{2k} P_\pi^{2k} + \sum_{2k+1 \geq 1} \gamma^{2k+1} (P_\pi)^{2k+1} \\ &= I + \sum_{2k \geq 2} \gamma^{2k} \left(\frac{v v^\top}{S-1} + e_S e_S^\top \right) + \sum_{2k+1 \geq 1} \gamma^{2k+1} P_\pi \\ &= I + \frac{\gamma^2}{1 - \gamma^2} \left(\frac{v v^\top}{S-1} + e_S e_S^\top \right) + \frac{\gamma}{1 - \gamma^2} P_\pi. \end{aligned}$$

Define $\eta := \frac{\gamma}{1 - \gamma^2}$. The non-zero singular values of $(I - \gamma P_\pi)^{-1}$ are the square roots of the eigenvalues of $A = (I - \gamma P_\pi)^{-1} ((I - \gamma P_\pi)^{-1})^\top$. We have

$$\begin{aligned} A &= (I - \gamma P_\pi)^{-1} \left((I - \gamma P_\pi)^{-1} \right)^\top = \left(I + \gamma \eta P_\pi^2 + \eta P_\pi \right) \left(I + \gamma \eta (P_\pi^2)^\top + \eta P_\pi^\top \right) \\ &= I + B, \end{aligned}$$

where $B := avv^\top + be_S e_S^\top + c(e_S v^\top + v e_S^\top)$ with $a = \frac{2\eta\gamma + \eta^2\gamma^2}{S-1} + \eta^2$, $b = 2\eta\gamma + \eta^2\gamma^2 + \frac{\eta^2}{S-1}$ and $c = (\eta + \eta^2\gamma)\frac{S}{S-1}$.

Moreover, if $\{\lambda_1, \dots, \lambda_k\}$ are the eigenvalues of B then the eigenvalues of A are $\{1 + \lambda_1, \dots, 1 + \lambda_k\}$.

Consider the basis $\{e_S, v\}$. For any a_1, a_2 ,

$$\begin{aligned} B(a_1 e_S + a_2 v) &= avv^\top(a_1 e_S + a_2 v) + be_S e_S^\top(a_1 e_S + a_2 v) \\ &\quad + c(e_S v^\top + v e_S^\top)(a_1 e_S + a_2 v) \\ &= a_1 a \langle v, e_S \rangle v + a_2 a \|v\|_2^2 v + a_1 b e_S + a_2 b \langle v, e_S \rangle e_S + c(a_1 \langle v, e_S \rangle e_S \\ &\quad + a_1 v + a_2 \|v\|_2^2 e_S + a_2 \langle v, e_S \rangle v) \\ &= (a_1 b + c a_1 \langle v, e_S \rangle + a_2 b \langle v, e_S \rangle + a_2 c \|v\|_2^2) e_S + (a_1 a \langle v, e_S \rangle + c a_1 \\ &\quad + a_2 \langle v, e_S \rangle + a_2 a \|v\|_2^2) v. \end{aligned}$$

Since $\|v\|_2^2 = S - 1$ and $\langle v, e_S \rangle = 0$, B has the representation in $\{e_S, v\}$ as:

$$\begin{aligned} \begin{bmatrix} b & c(S-1) \\ c & a(S-1) \end{bmatrix} &= \begin{bmatrix} 2\eta\gamma + \eta^2\gamma^2 + \frac{\eta^2}{S-1} & (\eta + \eta^2\gamma)S \\ (\eta + \eta^2\gamma)\frac{S}{S-1} & 2\eta\gamma + \eta^2\gamma^2 + \eta^2(S-1) \end{bmatrix} \\ &= \begin{bmatrix} \frac{\eta^2}{S-1} & (\eta + \eta^2\gamma)S \\ (\eta + \eta^2\gamma)\frac{S}{S-1} & \eta^2(S-1) \end{bmatrix} + (2\eta\gamma + \eta^2\gamma^2)I \\ &= C + (2\eta\gamma + \eta^2\gamma^2)I \end{aligned}$$

Hence, the eigenvalues λ_\pm of C are given by

$$\frac{1}{2} \left(\eta^2 \left((S-1) + \frac{1}{S-1} \right) \pm \sqrt{\eta^4 \left((S-1) + \frac{1}{S-1} \right)^2 + 4(\eta + \eta^2\gamma)^2 \frac{S^2}{S-1} - 4\eta^4} \right).$$

The non-zero singular values of $(I - \gamma P_\pi)^{-1}$ are thus 1 with multiplicity $S - 2$ and $\sqrt{\lambda_\pm + 2\eta\gamma + \eta^2\gamma^2 + 1}$. For $\gamma = 0.99$ and $S = 400$, we can check numerically that the two extreme singular values are equal to 996 and 0.05 respectively which matches the spectrum obtained for the Star graph in Figure 4.2.

4.C Empirical Evaluation: Additional Details

4.C.1 Graphical Structures

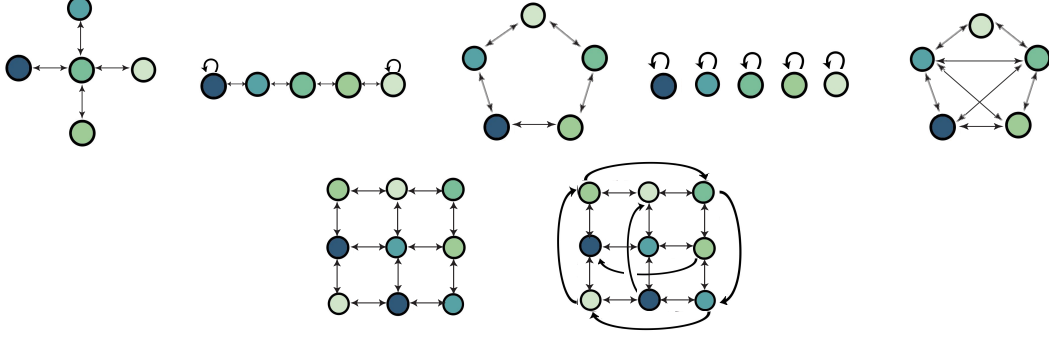


Figure 4.6: Different graphical structures with $S = 5$ states from left to right, Star, Chain, Torus1d, Disconnected, Fullyconnected (top). Two-dimensional graphical structures with $S = 9$ states: from left to right, Openroom and Torus2d (bottom).

In this section, we study the generalization characteristics of the representations induced by the SVD of the successor representation for several environment transition structures. We illustrate the different graphs over which we define a random walk, studied in Subsection 4.4.2 as well as some new ones, in Figure 4.6.

Our experiment consists in evaluating the value function on these different transition structures when $S = 400$ states. We consider three different reward vectors $r_\pi \in \mathbb{R}^S$: the all ones vector, the one-hot feature vector e_S , and a vector whose entries are drawn from zero-mean Gaussian distribution and normalized such that $\|r_\pi\|_\infty = 1$. We then sampled a dataset D of $n = 300$ pairs (s_i, y_i) where we performed a Monte Carlo rollout to obtain the returns $(y_i)_{i=1}^n$. The targets are the value functions induced by the random walk.

We are interested in comparing our generalization bound to the empirical excess risk on these domains. Our bound looks at the regime $n \geq d_{\text{eff}}(F_k)$. We choose $k \leq \frac{n}{2}$ as an heuristic way of achieving this. We report in Figure 4.7 the approximation error (Figure 4.7 Left), the empirical excess risk (Figure 4.7 Middle) and the theoretical excess risk (Figure 4.7 Right) obtained when using the representation $\phi = F_k$ on these different graph structures.

Star: Baird’s star graph [Baird, 1995] consists in $S - 1$ states which are the star corners and a state S which is the star center. A random walk on this star graph induces a transition function such that all star corners transition to the star center and the star center goes to the star corners. There are two extreme cases in terms of rewards: either the reward is the same for all $s_i, i \neq S$, (e.g. the all ones reward vector or the one-hot vector e_S) or not. If the reward is the same for all $s_i, i \neq S$, then this is effectively a 2 state structure, so we only really need 1 feature to distinguish between the value of the star corners and the value of the star center. However, if the reward is different for all $s_i(i \neq S)$ then we effectively have $(S - 1)$ tuples (s_i, s_S) which can be thought of as independent graphical structures and we thus expect to need all the features to distinguish between their values. We can see this in Figure 4.7 that for the all ones reward vector and the one-hot reward vector e_S , the error with $k = 1$ is very good but for the Gaussian reward, the error with $k = 1$ is high.

Chain: This is a S -state connected graph with 2 pendant states and $(n - 2)$ states of degree two. The shapes of the curves are similar to the Torus1d but we can notice that the errors are larger for each feature dimension k . This is intuitive as for instance in the case of an all ones reward vector, the values are not the same for each state due to the two end states of the chain, implying that more than one feature is needed to generalize the value function.

Openroom: This is a two-dimensional grid with S states. States strictly inside the grid have four neighbours. States belonging to one (reps. two) edges are of degree three (resp. two). As we observed in Figure 4.2, the Openroom domain does not generalize as well as the Torus2d which can be explained by their difference in effective dimension.

Torus1d: This is a wrap-around version of the Chain. State i transitions to state $(i + 1) \bmod S$ and state $(i - 1) \bmod S$. We can see that the curve showing the empirical excess risk (Middle) corresponding to the Gaussian reward vector has a sweet spot which is also predicted by our theory. Moreover, when all states have the same reward, their values are identical. Hence, in that case, only one

feature is enough to have very low error which is shown both empirically and by our theoretical bound on Figure 4.7.

Torus2d: It is a wrap-around version of the Openroom domain such that each state has four different neighbors. We can see in Figure 4.2 that the Torus1d and Torus2d have similar effective dimension but the decay of the singular values is faster in the case of Torus2d translating into smaller approximation errors in Figure 4.7 (Middle). This results in overall lower excess risk for the Torus2d indicating it generalizes in general better than its one-dimensional counterpart. Just like for the Torus1d, in the case of the Gaussian reward vector, there is a non trivial optimal number of features k minimizing the excess risk, which we can notice is smaller than for the Torus1d.

Disconnected: This graph consists of S states that self-transition. We do not expect the successor representation to generalize well within this MDP as we cannot leverage knowledge from one feature state to another. This idea was already captured by the effective dimension shown in Figure 4.2. The plots in Figure 4.7 corroborates this both empirically and theoretically showing that its excess risk is indeed the highest across all transition structures considered.

Fullyconnected: This is a connected graph of S states where each state can transition to $(S - 1)$ states. The first singular vector, which is the constant vector, is very good in terms of effective dimension but the second vector has high effective dimension. When the rewards are the same in each state, their values are identical. In that case, one feature is enough to distinguish between the S states leading to good generalization in that case. Additional features must be misleading as the excess risks rises significantly from a number of features $k = 2$.

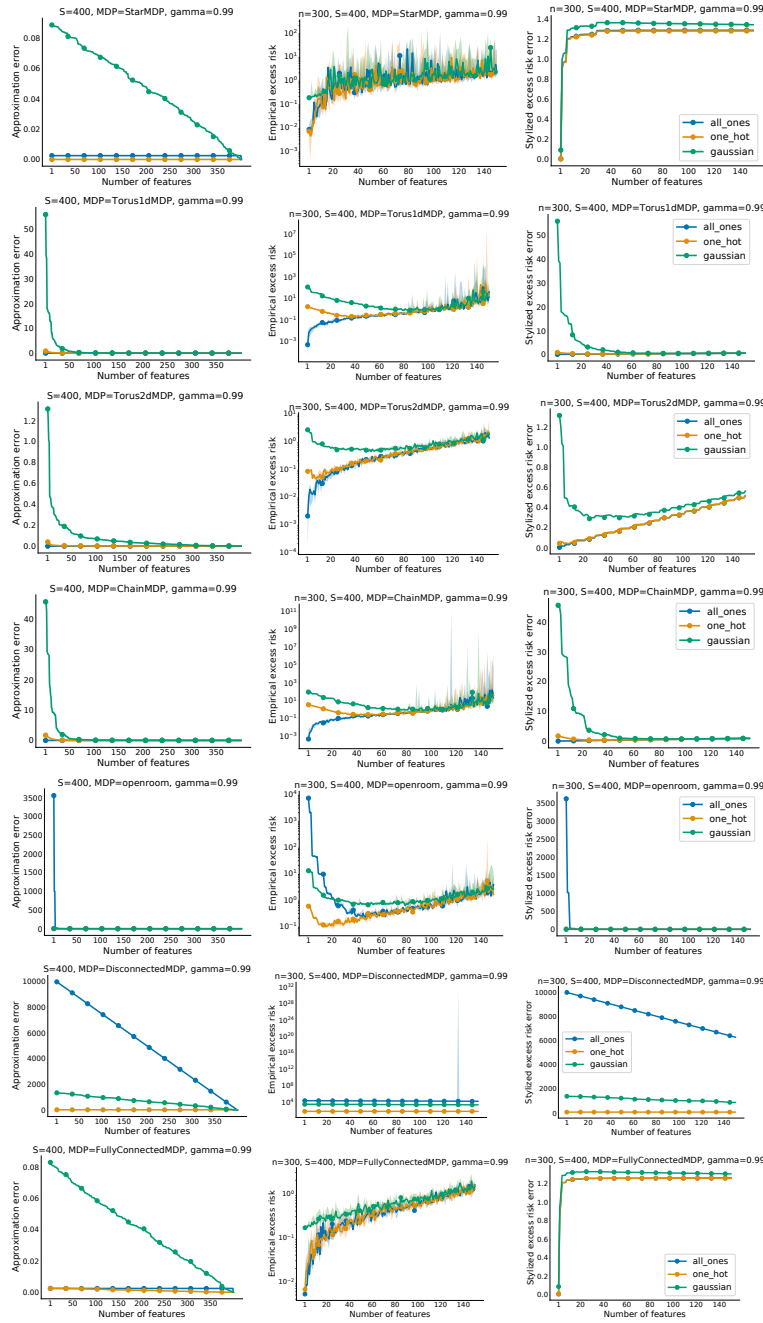


Figure 4.7: Approximation error $\|P_{F_k} V^\pi\|$ given a one-hot, all-ones and Gaussian reward vector and for MDPs with different graphical structures (left). Median empirical excess risk $\mathcal{E}(V_{F_k, \hat{w}})$ given a one-hot, all-ones and Gaussian reward vector (middle). Theoretical excess risk for a representation $\Phi_k = F_k$ and a one-hot, all-ones and Gaussian reward vector (right). The median is over 5 random seeds and shading gives 95% confidence intervals.

4.C.2 Full Atari Results

For all experiments, we used the hyperparameters provided by Dopamine [Castro et al., 2018].

Compute. For our experiments on Atari, we used Tesla V100 GPUs and P100 for all runs. To obtain the pretrained deep representations for each deep RL agent, we ran a total of 5 runs / game \times 60 games / algorithm \times 5 algorithms = 1500 runs. Each of these runs takes around 5 days. Additionally, for the auxiliary loss experiment, we ran a total of 5 runs / game \times 5 games / algorithm \times 2 algorithms = 50 runs. In this setting, each run takes around 1 day. Overall, the amount of compute is of 7050 days of GPU training.

We provide a per-game comparison of the effective dimension of the representations induced by DQN, DQN (Adam), Rainbow, IQN and M-IQN throughout training in Figure 4.9 for all 60 Atari games in the online setting to complement the results presented in Figure 4.4 in the main part of the paper.

For the offline experiment presented in Figure 4.5, we use the same mini-batch sampled for the temporal-difference loss \mathcal{L}_{TD} for computing the auxiliary loss \mathcal{L}_{ϕ} . Our combined loss is then $\mathcal{L}_{\alpha} = (1 - \alpha)\mathcal{L}_{\text{TD}} + \alpha\mathcal{L}_{\phi}$. We ran a hyperparameter sweep over α on the five games displayed in Figure 4.8 and found that a value of $\alpha = 0.1$ worked well. We provide per-game training curves for IQN agents for 17 Atari games in Figure 4.10 as well as the effective dimension (see Figure 4.11) of their induced representations computed with a batch size of 2^{15} . We also complement these results with the rank of these representations as a function of training in Figure 4.12 and Figure 4.13 as a proxy for the approximation error.

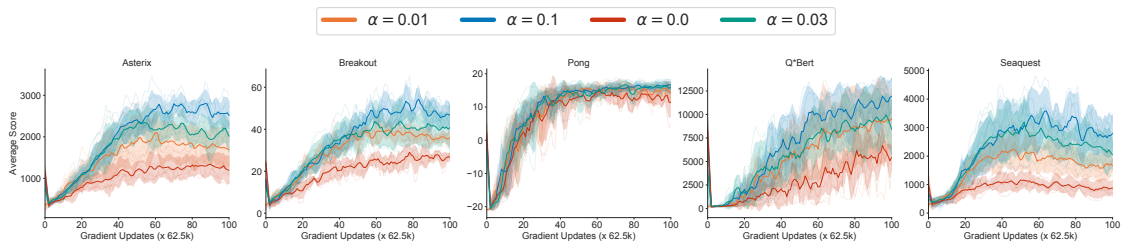


Figure 4.8: Sweeping over various values of α when adding the auxiliary loss \mathcal{L}_{ϕ} to IQN.

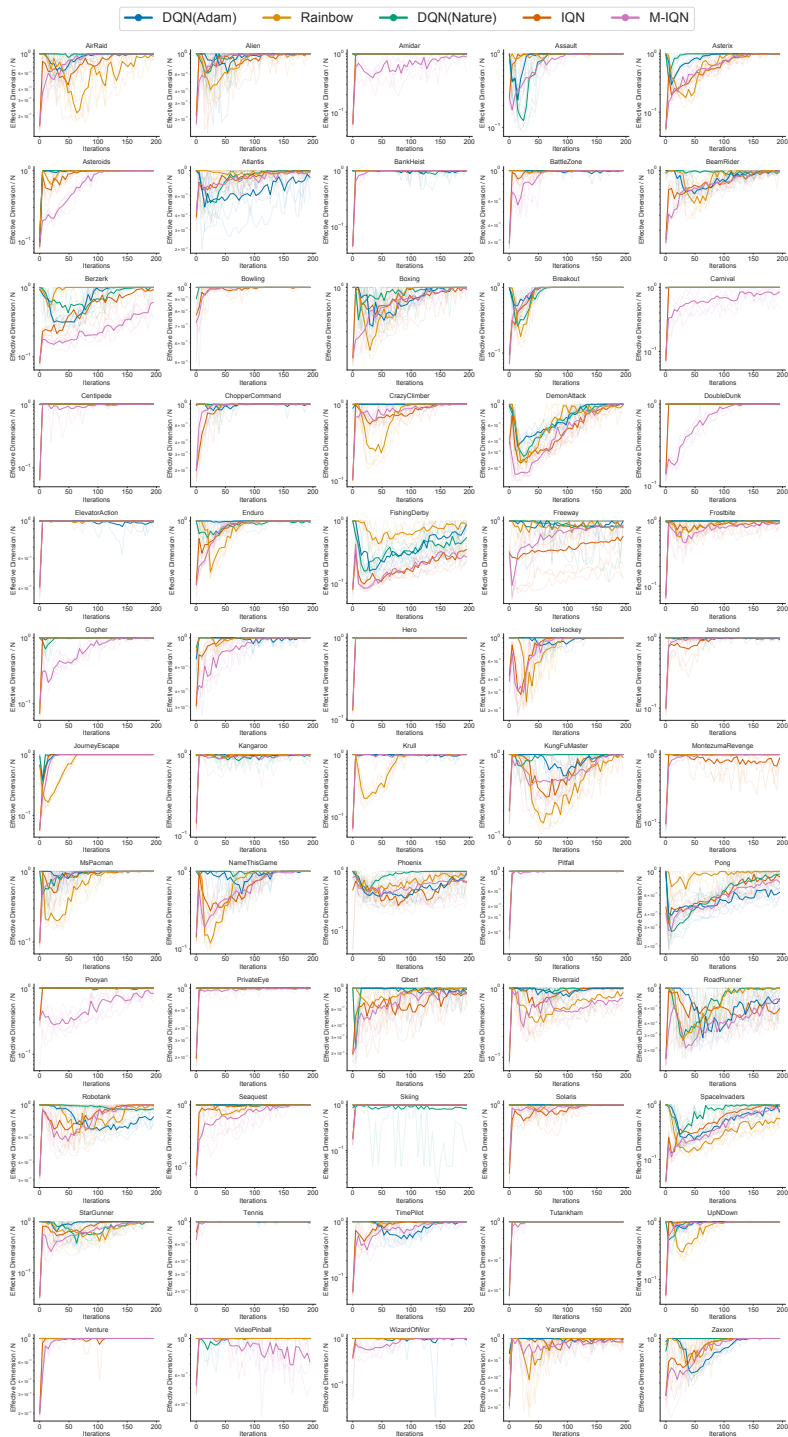


Figure 4.9: Average estimate (darker color) of the effective dimension normalized by the batch size used $N = 2^{15}$ on DQN(Nature), DQN(Adam), Rainbow, IQN and M-IQN on all 60 Atari games computed using 5 independent runs. Individual runs are shown with a lighter color.

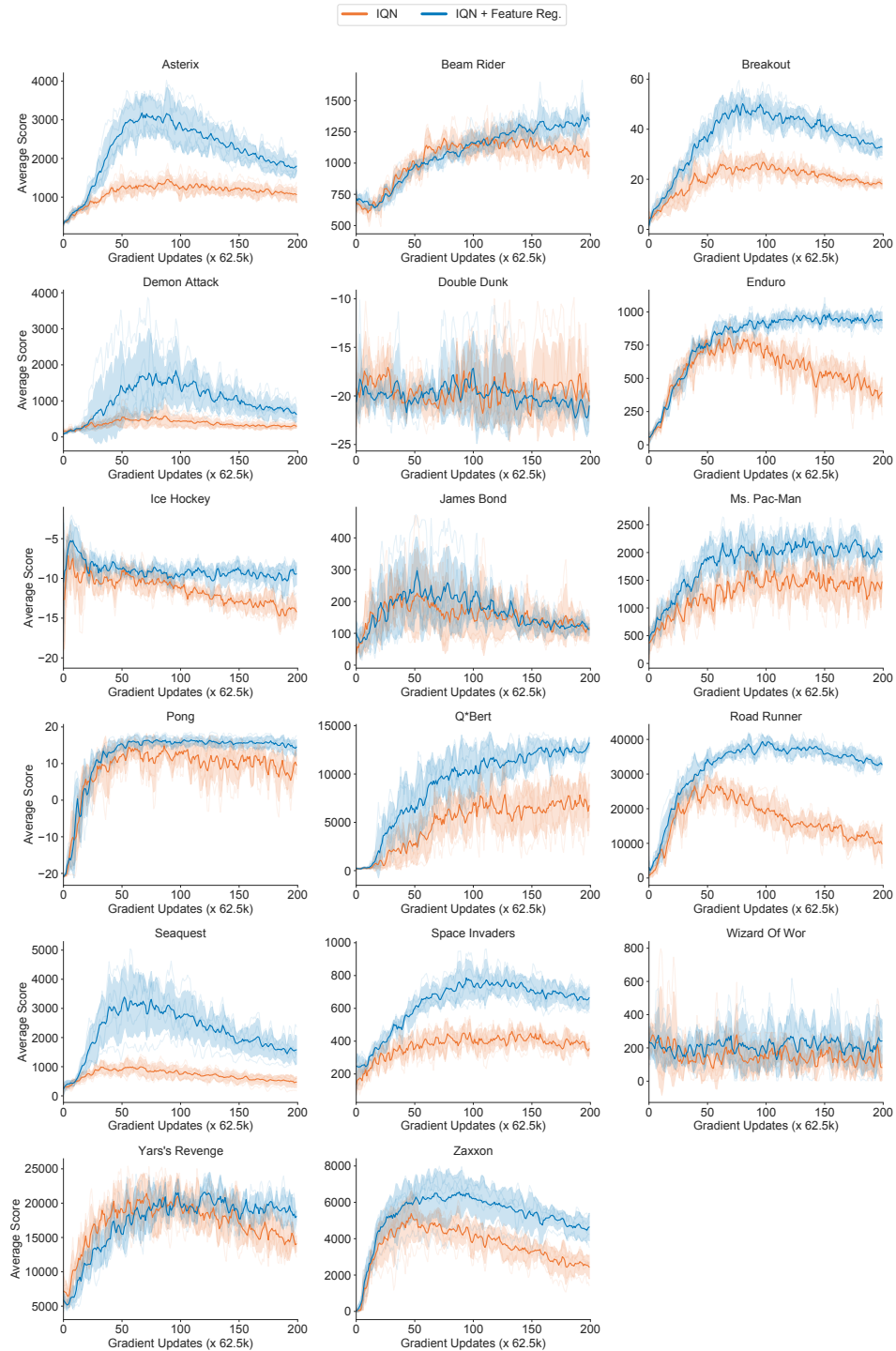


Figure 4.10: Per-game learning curves of IQN and IQN with feature regularization L_ϕ on 17 Atari games in the offline RL setting.

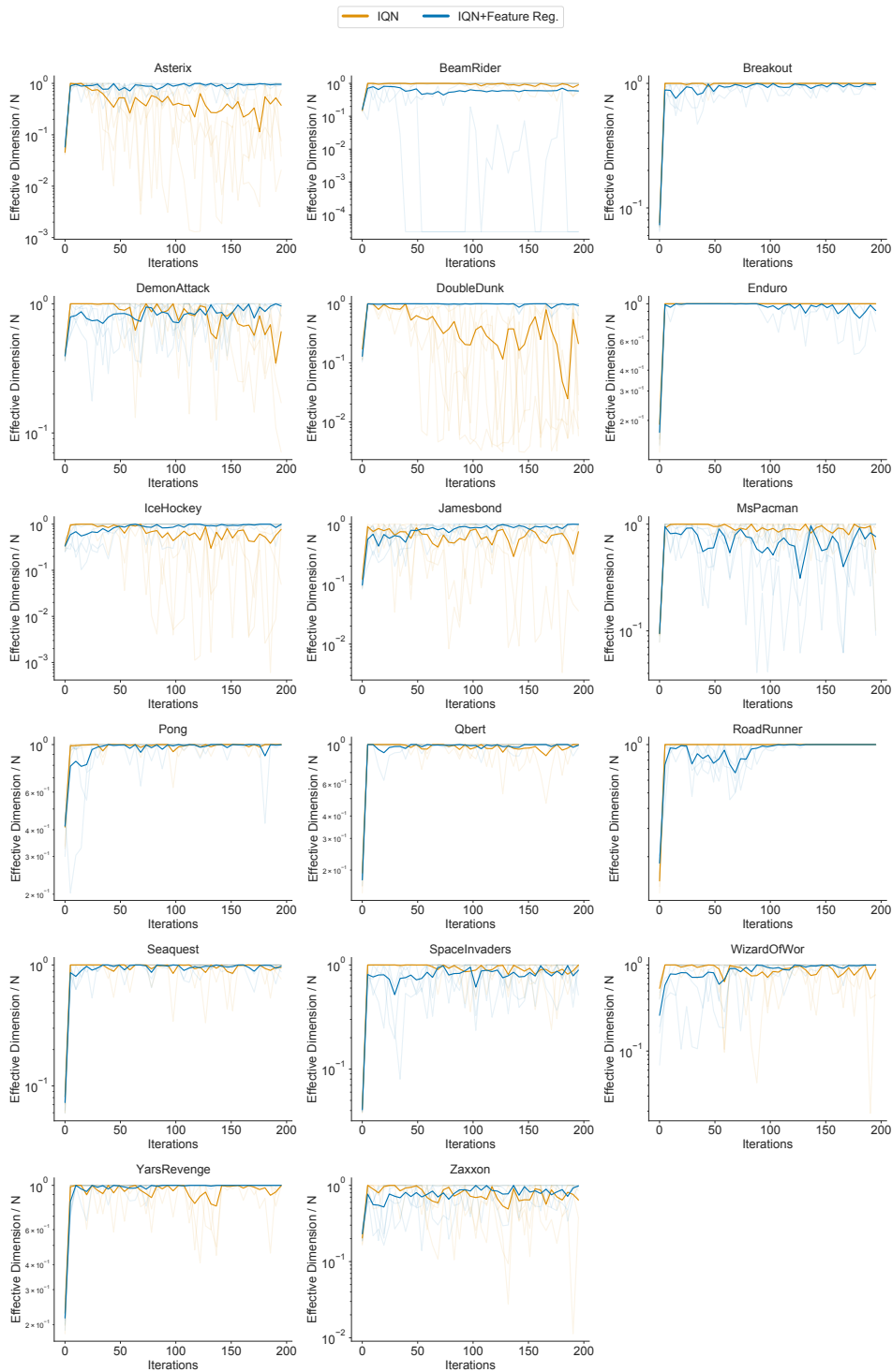


Figure 4.11: Per-game effective dimension normalized by the batch size $N = 2^{15}$ of IQN and IQN with feature regularization L_ϕ on 17 Atari games in the offline RL setting, using 5 independent runs. Individual runs are shown with a lighter color.

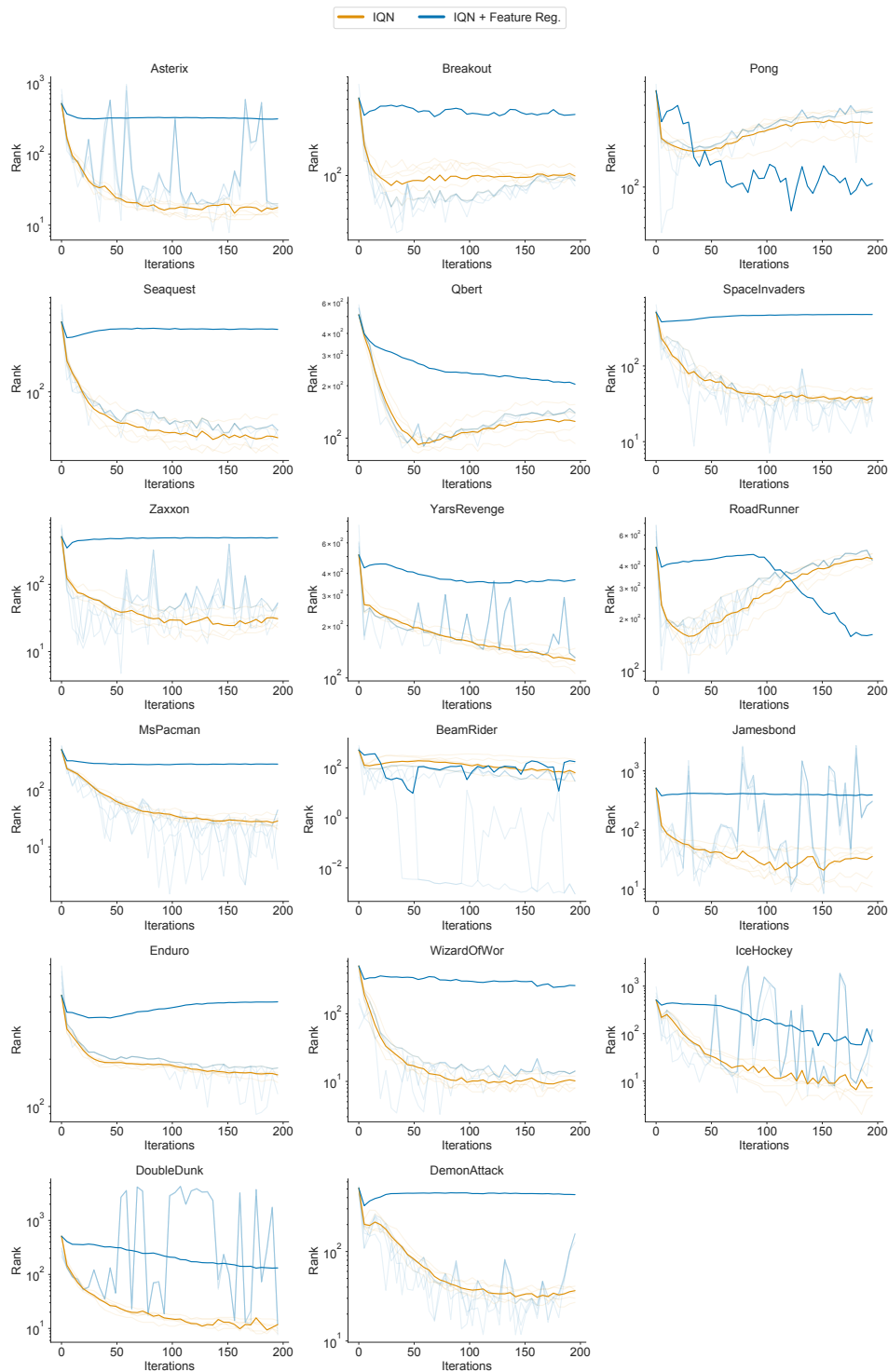


Figure 4.12: Per-game rank of IQN and IQN with feature regularization L_ϕ computed with a batch size $N = 2^{15}$ on 17 Atari games in the offline RL setting, using 5 independent runs. Individual runs are shown with a lighter color.

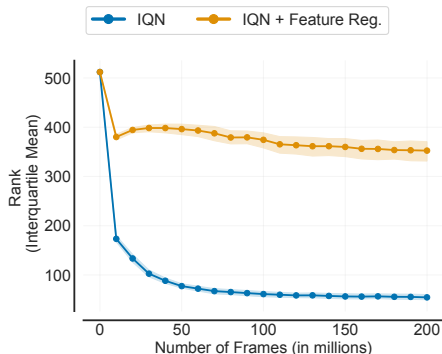


Figure 4.13: Interquartile mean (IQM) [Agarwal et al., 2021b] for the rank of representations induced by IQN and IQN with feature regularization L_ϕ computed with a batch size $N = 2^{15}$ on 17 Atari games in the offline setting.

4.D Societal Impact

This paper contributes to the fundamental understanding of state representations, characterizing their generalization capacity. Our work suggests that algorithms making use of representations minimized by the excess risk bound from Theorem 6 can improve their performance. However, when making the choice of such a representation, we did not focus on other important factors like the computational cost of learning these representations, their scalability or the biases these representations can propagate resulting into possible discriminatory outcomes or dangerous behaviours. We suggest that practitioners should not only consider our generalization characterization of representations but also ethical deliberations.

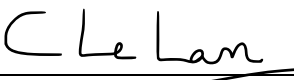
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor (**only required where there isn't already a statement of contribution within the paper itself**).

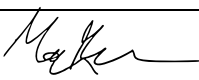
Title of Paper	On the Generalization of Representations in Reinforcement Learning
Publication Status	<input checked="" type="checkbox"/> Published <input type="checkbox"/> Accepted for Publication <input type="checkbox"/> Submitted for Publication <input type="checkbox"/> Unpublished and unsubmitted work written in a manuscript style
Publication Details	Charline Le Lan, Stephen Tu, Adam Oberman, Rishabh Agarwal, Marc G. Bellemare. On the Generalization of Representations in Reinforcement Learning. <i>In International Conference on Artificial Intelligence and Statistics (AISTATS) 2022</i>

Student Confirmation

Student Name:	Charline Le Lan		
Contribution to the Paper	I led the project, wrote large parts of the paper, proved some theoretical results, implemented and ran experiments on graphs and the four-room domain, generated the deep RL plots for the paper. Stephen proved some theoretical results and contributed to the writing of theoretical sections. Adam took part in some discussions during which we defined the project with all other authors. Rishabh wrote code and ran some deep RL experiments. Marc advised the project, contributed to writing and provided feedback on the paper. All authors contributed to the development of the project through discussions and all authors reviewed the manuscript.		
Signature		Date	March 23, 2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Marc G. Bellemare			
Supervisor comments	With this work Charline demonstrated a strong aptitude at integrating technical elements from different fields, and changed the way we think about generalization in RL.		
Signature		Date	13/04/23

This completed form should be included in the thesis, at the end of the relevant chapter.

5

A Novel Stochastic Gradient Descent Algorithm for Learning Principal Subspaces

Abstract

Many machine learning problems encode their data as a matrix with a possibly very large number of rows and columns. In several applications like neuroscience, image compression or deep reinforcement learning, the principal subspace of such a matrix provides a useful, low-dimensional representation of individual data. Here, we are interested in determining the d -dimensional principal subspace of a given matrix from sample entries, i.e. from small random submatrices. Although a number of sample-based methods exist for this problem (e.g. Oja's rule [Oja, 1982]), these assume access to full columns of the matrix or particular matrix structure such as symmetry and cannot be combined as-is with neural networks [Baldi and Hornik, 1989]. In this paper, we derive an algorithm that learns a principal subspace from sample entries, can be applied when the approximate subspace is represented by a neural network, and hence can be scaled to datasets with an effectively infinite number of rows and columns. Our method consists in defining a loss function whose minimizer is the desired principal subspace, and constructing a gradient estimate of this loss whose bias can be controlled. We complement our theoretical analysis with a series of experiments on synthetic matrices, the MNIST dataset [LeCun et al., 2010] and the reinforcement learning domain PuddleWorld [Sutton, 1995] demonstrating the usefulness of our approach.

5.1 Introduction

Learning compact representations of data while minimizing information loss is at the heart of machine learning. A common approach for doing so is to learn a d -dimensional principal subspace that explains most of the variation in the data, what is known as principal component analysis (PCA). For small datasets, PCA can be accomplished by computing the singular value decomposition of the relevant data matrix. For sufficiently large datasets, however, this approach is impractical and one must instead turn to a stochastic or sample-based procedure.

Streaming PCA algorithms learn an approximate principal subspace by sampling columns from the data matrix Ψ and performing an incremental update that moves their approximation closer to the true subspace [e.g. Krasulina, 1970, Oja, 1982, Gemp et al., 2021, 2022]. Central to these methods is the computation of the inner product between a full matrix column and the approximate subspace as well as a step to normalize the basis vectors parametrizing this subspace, making these methods most suited to problems where there are relatively few matrix rows. Another line of work learns the principal subspace as the by-product of a low-rank linear regression problem. In this case, the learner forms a product Φw_t where Φ encodes the approximate subspace and w_t is a per-column weight vector; the aim is to minimize the Euclidean distance between Φw_t and the column Ψ_t [Srebro and Jaakkola, 2003, Jin et al., 2016, Sun and Luo, 2016]. This approach has been effective for learning state representations in reinforcement learning [Bellemare et al., 2019, Gelada et al., 2019, Dabney et al., 2021, Lyle et al., 2021], but can only handle a small number of columns, owing to the need to store an explicit weight vector for each.

In this paper, we consider the problem of learning a d -dimensional principal subspace by means of a neural network. Following common usage, we view the neural network as a mapping from the original input space to a d -dimensional vector space. We propose a fully sample-based algorithm which exhibits the best of the two classes of approaches above. Rather than maintain the weight vector w_t in memory, we instead estimate it on-the-fly from samples – effectively making the

weight vector implicit. We use the weight vector estimate to construct a gradient of a suitable loss function, on which we perform stochastic gradient descent in order to determine an approximation to the d -dimensional principal subspace. Key to our approach is the derivation of the gradient in terms of Danskin’s theorem. Although the naive plug-in gradient fails to be an unbiased estimate and can perform quite poorly in practice, an unbiased estimate is obtained by constructing two independent weight vector estimates. These estimates are derived from a technique known as the LISSA (Linear (time) Stochastic Second-Order Algorithm, see Agarwal et al. [2017]) that we apply to produce a sequence of asymptotically-unbiased estimators of the inverse covariance matrix $(\Phi^\top \Phi)^\dagger$. Based on its origins, we call the result the *Danskin-LISSA* algorithm.

In Section 5.5, we show that our algorithm can recover the principal subspace of synthetic matrices and of MNIST images, while only observing a small subset of the data matrix at each update. We further demonstrate the effectiveness of our method for representation learning in reinforcement learning, specifically by learning a neural network-based approximation to the principal subspace of the successor measure [Blier et al., 2021] in the Puddle World domain [Sutton, 1995].

5.2 Background

5.2.1 Problem Statement

We consider a collection of column functions $\{\psi_t \in \mathbb{R}^S\}_{t \in \mathcal{T}}$ where \mathcal{T} is an index set, and where each function ψ_t maps row indices to real values. For instance, ψ_t can be one observation or one data point from the data matrix Ψ . We assume that the column indices and the row indices are drawn i.i.d from a distribution λ on \mathcal{T} and ξ on \mathcal{S} respectively ¹. For a given integer $d \in \mathbb{N}$ and a *row representation* $\phi : \mathcal{S} \rightarrow \mathbb{R}^d$, we define the *representation loss*

$$\mathcal{L}(\phi) = \mathbb{E}_{t \sim \lambda} \left[\min_{w_t \in \mathbb{R}^d} \mathbb{E}_{s \sim \xi} \left[(\phi(s)^\top w_t - \psi_t(s))^2 \right] \right]. \quad (5.1)$$

¹We assume that $\xi(s) > 0$ for all row indices $s \in \mathcal{S}$ and that $\lambda(t) > 0$ for all column indices $t \in \mathcal{T}$.

The representation loss describes the approximation error incurred by fitting the column function ψ_t with the d -dimensional linear approximation $\phi(s)^\top w_t$, on average over draws from λ . Here, we are interested in determining a d -dimensional representation ϕ that minimises $\mathcal{L}(\phi)$ among all such representations.

For now, let us consider the case in which \mathcal{S} and \mathcal{T} are of finite sizes S and T , respectively. In this case, we may write $\Phi \in \mathbb{R}^{S \times d}$ for the *feature matrix* whose rows are $(\phi(s))_{s \in \mathcal{S}}$ and $\Psi \in \mathbb{R}^{S \times T}$ for the data matrix whose columns are $(\psi_t)_{t \in \mathcal{T}}$. If additionally $W \in \mathbb{R}^{d \times T}$ is a weight matrix, then finding the function ϕ that minimizes Equation (5.1) is equivalent to jointly minimizing the loss $\mathcal{L}(\Phi, W)$ over Φ and W , where

$$\mathcal{L}(\Phi, W) = \|\Xi^{1/2}(\Phi W - \Psi)\Lambda^{1/2}\|_F^2. \quad (5.2)$$

Here, $\Xi \in \mathbb{R}^{S \times S}$ (resp. $\Lambda \in \mathbb{R}^{T \times T}$) is a diagonal matrix with entries $\{\xi(s) : s \in \mathcal{S}\}$ (resp. $\{\lambda(t) : t \in \mathcal{T}\}$) on the diagonal. For a given Φ , we write

$$W_\Phi^* \in \arg \min_{W \in \mathbb{R}^{d \times T}} \mathcal{L}(\Phi, W) \quad \mathcal{L}(\phi) = \mathcal{L}(\Phi, W_\Phi^*). \quad (5.3)$$

From standard linear algebra (see Lemma 14 in Appendix 5.A), in closed form we have

$$W_\Phi^* = (\Phi^\top \Xi \Phi)^\dagger \Phi^\top \Xi \Psi. \quad (5.4)$$

Note that this expression does not depend on the column distribution Λ . We will use this matrix form to derive a gradient-based algorithm in the next section.

Equation (5.2) describes a weighted low-rank approximation problem [Srebro and Jaakkola, 2003]. Its solutions are the set of matrices Φ whose columns span the d -dimensional subspace of left singular vectors of Ψ with respect to the inner product $(x, y)_\Xi = x^\top \Xi y$ (see Proposition 1 in Appendix 5.A for a proof). If in addition the columns of Ψ have mean zero, this corresponds to determining the subspace spanned by the d principal components of Ψ . Consequently, in the finite case our objective is to find a state representation whose implied feature matrix has

columns that span this subspace. As we will see, one advantage of this objective over the more usual Rayleigh quotient in the case $d = 1$,

$$\mathbb{E}_{s, s' \sim \xi, t \sim \lambda} [\phi(s)\psi_t(s)\psi_t(s')\phi(s')],$$

is that its gradient incorporates an error term $\mathbb{E}_{s \sim \xi, t \sim \lambda} [(\phi(s)^\top w_t - \psi_t(s))w_t^\top]$ that is naturally zero at a minimizer.

5.3 PCA from Samples

We assume access to a model from which we may repeatedly sample row indices according to the distribution ξ and the values taken on at those row indices by column functions sampled from λ . We are interested in the setting in which it is undesirable or impossible to sample the entire collection of column functions for a given state, or an entire column function all at once. This is different from the setting that approaches such as Oja's method [Oja, 1982] or the recent EigenGame [Gemp et al., 2021] have considered for their experiments, which in matrix terms assume that it is possible to sample entire rows or columns from Ψ (for a longer discussion on prior work, see Section 5.4).

Let us begin by expressing the gradient of the loss function $\mathcal{L}(\Phi, W)$. In matrix form, this is

$$\nabla_{\Phi} \mathcal{L}(\Phi, W) = 2\Xi(\Phi W - \Psi)\Lambda W^\top \quad (5.5)$$

$$\nabla_W \mathcal{L}(\Phi, W) = 2\Phi^\top \Xi(\Phi W - \Psi)\Lambda \quad (5.6)$$

When the number of columns T is small, finding an optimal ϕ can be accomplished by optimizing the loss function $\mathcal{L}(\Phi, W)$ using a nested or two-timescale optimization procedure based on unbiased estimates of these gradients. For example, the pair of update rules

$$\begin{aligned} \phi(s) &\leftarrow \phi(s) - \alpha(\phi(s)^\top w_t - \psi_i(s))w_t \\ w_t &\leftarrow w_t - \beta\phi(s)(\phi(s)^\top w_t - \psi_i(s)) \end{aligned} \quad (5.7)$$

finds an optimal representation ϕ under suitable conditions on the step-sizes α and β . This is because the loss $\mathcal{L}(\Phi, W)$ is convex in W when Φ is fixed and the two-timescale algorithm allows us to approximately run gradient descent on the objective we care about.

When T is large (or infinite), however, it may be expensive (or impossible) to store a separate weight vector for each column. Instead, we rely on a form of the gradient of the loss $\mathcal{L}(\phi)$ in which the weight vector is implicit.

Lemma 12. *Let $\beta > 0$ be a regularization parameter. The loss $\mathcal{L} : \mathbb{R}^{S \times d} \rightarrow \mathbb{R}$ defined by*

$$\mathcal{L}(\Phi) = \min_{W \in \mathbb{R}^{d \times T}} \left(\|\Xi^{1/2}(\Phi W - \Psi)\Lambda^{1/2}\|_F^2 + \beta \|W\|_F^2 \right) \quad (5.8)$$

is continuously differentiable, with gradient

$$\nabla_{\Phi} \mathcal{L}(\Phi) = 2\Xi(\Phi W_{\Phi}^* - \Psi)\Lambda W_{\Phi}^{*\top},$$

where

$$W_{\Phi}^* = (\Phi^{\top} \Xi \Phi + \beta I)^{-1} \Phi^{\top} \Xi \Psi. \quad (5.9)$$

Proof. The proof is similar to the one of Danskin's theorem [Danskin, 2012]. By linear algebra, the unique minimizer W_{Φ}^* in Equation (5.8) is given by Equation (5.9), which is itself differentiable with respect to Φ . By the chain rule, we have

$$\nabla_{\Phi} \mathcal{L}(\Phi) = \nabla_{\Phi} \mathcal{L}(\Phi, W_{\Phi}^*) + \left(\frac{\partial W_{\Phi}^*}{\partial \Phi} \right)^{\top} \frac{\partial}{\partial W_{\Phi}^*} \mathcal{L}(\Phi, W_{\Phi}^*).$$

Now, since W_{Φ}^* is defined as the (unconstrained) minimizer of $\mathcal{L}(\Phi, W_{\Phi}^*)$, its gradient with respect to the second argument vanishes at W_{Φ}^* , and so second term is zero. The result then follows from the definition of $\nabla_{\Phi} \mathcal{L}(\Phi, W)$ in Equation (5.5). \square

The idea is to use an instantaneous estimate of W_{Φ}^* to update the row representation in the negative direction of the (estimated) gradient of $\mathcal{L}(\phi)$. As we will see, such an estimate can be obtained by sampling as little as a single column

and a small number of rows. In effect, given a sample row index s our goal is to obtain a gradient estimate $\hat{g}(s)$ such that

$$\phi(s) \leftarrow \phi(s) - \alpha \hat{g}(s) \quad (5.10)$$

should converge to an optimal representation under suitable conditions on the time-varying step-size α . In Subsection 5.5.3, we will discuss how Equation (5.10) can be applied to learn parametrized row representations such as those described by neural networks.

Before describing our approach, it is worth noting that the procedure that naively estimates W_{Φ}^* from a subset of rows and columns results in a biased gradient estimate. That is, suppose we are given the sample row indices s, s', s_1, \dots, s_n and sample column t . If we write $\hat{\Phi}$ for the matrix whose rows are $\phi(s_1), \dots, \phi(s_n)$ and construct the empirical covariance matrix $\hat{C} = \hat{\Phi}^\top \hat{\Phi}$, then we find that the estimate

$$\hat{g}_{\text{NAIVE}}(s) = \hat{w}_t (\phi(s)^\top \hat{w}_t - \psi_t(s)) \quad \hat{w}_t = \hat{C}^\dagger \phi(s') \psi_t(s') \quad (5.11)$$

is not an unbiased estimate of $\nabla_{\phi(s)} \mathcal{L}(\Phi)$ because each factor is not independent from each other. In fact, the bias can be quite substantial when n is small, as we empirically show in Section 5.5.

5.3.1 An Improved Gradient Estimate

One issue with the estimate of Equation (5.11) is that the estimated weight vector \hat{w}_t is itself a largely biased estimate of the optimal weight vector for column t (that is, the t^{th} column of W_{Φ}^* , $W_{\Phi,t}^*$). Conversely, unbiasedness is obtained if \hat{w}_t satisfies

$$\mathbb{E}[\hat{w}_t] = W_{\Phi,t}^*,$$

and if the term \hat{w}_t^\top is an independent, also unbiased estimate of $W_{\Phi,t}^{*\top}$ in Lemma 12. To reduce the bias of the naive estimate, we will construct two low-biased estimates of the inverse covariance matrix $(\Phi^\top \Phi)^\dagger$, \hat{C} and \hat{C}' , from which we derive two independent weight estimates \hat{w}_t and \hat{w}'_t .

Before we explain how to obtain these estimates, let us describe our algorithm at a high level. We begin by drawing three row indices s, s', s'' and a column index t . We then construct the weight estimates

$$\hat{w}_t = \hat{C}\phi(s')\psi_t(s') \quad \hat{w}'_t = \hat{C}'\phi(s'')\psi_t(s''),$$

and then the gradient estimate

$$\hat{g}_{\text{DL}}(s) = \hat{w}'_t \left(\phi(s)^\top \hat{w}_t - \psi_t(s) \right). \quad (5.12)$$

which uses two LISSA estimators [Agarwal et al., 2017] to construct independent weight estimates by application of Danskin's theorem. In effect, using two separate weight estimates effectively allows us to estimate the outer product $W_{\Phi,t}^* \left(W_{\Phi,t}^* \right)^\top$ appearing in Lemma 12 with a very low bias and hence obtain a gradient estimate that is overall low-biased, up to a multiplicative factor that we fold into the step-size parameter.

Theorem 8. *Let $e_s \in \mathbb{R}^S$ denote a basis vector. Given two independent unbiased estimates \hat{C} and \hat{C}' of the inverse covariance, for $s \sim \xi$, the gradient estimate $\hat{g}_{\text{DL}}(s)$ given in Equation (5.12) satisfies*

$$\mathbb{E}[e_s \hat{g}_{\text{DL}}(s)^\top] = \Xi(\Phi W_\Phi^* - \Psi)\Lambda W_\Phi^{*\top}.$$

Note that the estimate $\hat{g}_{\text{DL}}(s)$ does not require the set of columns \mathcal{T} to be finite. As such, our procedure can also be used to learn the principal components of infinite sets of columns; we will demonstrate this point in Subsection 5.5.3.

5.3.2 Estimate of the Weight Vector $W_{\Phi,t}^*$

We begin by deriving a procedure which, given access to a stream of sample row representations $\left(\phi(s_j) \right)_{j=1}^\infty$, asymptotically produces an unbiased estimate of the optimal weight vector for a given column t .

Central to our procedure is an estimate \hat{C} of the inverse covariance matrix $(\Phi^\top \Xi \Phi)^\dagger$. We construct this estimate by embedding what is known as the *LISSA estimator* [Agarwal et al., 2017, originally used to estimate the Hessian inverse].

Our algorithm is parameterised by two scalars, κ and J , which trade off estimator variance with sample complexity. All proofs can be found in Appendix 5.B.

To begin, consider an arbitrary matrix $\Phi \in \mathbb{R}^{S \times d}$ and denote $\|\cdot\|_{\text{op}}$ the spectral norm. For any $0 < \kappa < 2\|\Phi^\top \Xi \Phi\|_{\text{op}}^{-1}$, the Moore-Penrose pseudo-inverse of $(\Phi^\top \Xi \Phi)^\dagger$ has a Neumann series expansion of the form

$$(\Phi^\top \Xi \Phi)^\dagger = \kappa \sum_{i=0}^{\infty} (I - \kappa \Phi^\top \Xi \Phi)^i. \quad (5.13)$$

Here, κ is a scaling parameter that ensures the convergence of the series. Denoting S_j the first j terms of the above series, we have that

$$S_j = \kappa I + (I - \kappa \Phi^\top \Xi \Phi) S_{j-1}.$$

We use this observation to build an estimator of $(\Phi^\top \Xi \Phi)^\dagger$ with access to a finite number of samples from \mathcal{S} .

Definition 12 (LISSA estimator). *Let $\Phi \in \mathbb{R}^{S \times d}$ be a feature matrix. Let $s_{1:J} = \{s_1, s_2, \dots, s_J\}$ be J i.i.d. row indices sampled from ξ . Let $\kappa_0 \in (0, 2)$ and $\kappa = \kappa_0 / \sup_{s_{1:J}} \|\phi(s_i)\|_2^2$. The j -LISSA estimator $\widehat{\Delta}_j$ is recursively given by*

$$\begin{aligned} \widehat{\Delta}_0 &= \kappa I \\ \widehat{\Delta}_j &= \kappa I + (I - \kappa \phi(s_j) \phi(s_j)^\top) \widehat{\Delta}_{j-1}, \quad 0 < j \leq J. \end{aligned} \quad (5.14)$$

Lemma 13 (Bias of LISSA). *For $0 < \kappa < \sup_{s_{1:J}} 2\|\phi(s_i)\|_2^{-2}$, the bias of $\widehat{\Delta}_j$ with respect to $(\Phi^\top \Xi \Phi)^\dagger$ is given by*

$$\mathbb{E}(\widehat{\Delta}_j) - (\Phi^\top \Xi \Phi)^\dagger = -(\Phi^\top \Xi \Phi)^\dagger (I - \kappa \Phi^\top \Xi \Phi)^{j+1}$$

In particular, this bias asymptotically vanishes, in the sense that

$$\lim_{j \rightarrow \infty} \mathbb{E}(\widehat{\Delta}_j) - (\Phi^\top \Xi \Phi)^\dagger = 0.$$

While for any finite value of J , the LISSA estimator $\widehat{\Delta}_j$ is not an unbiased estimate, Lemma 13 establishes that its bias can be made arbitrarily small with enough samples. In our experiments, we will show that with few row samples this results in substantially better convergence compared to a naive estimate of the covariance matrix.

In Definition 12, the parameter κ controls the rate of convergence of the full Neumann series: larger values of κ result in faster convergence, requiring fewer samples to obtain an estimate that has little bias with regards to the inverse covariance matrix. However, larger values of κ (κ is bounded above as per Definition 12) also produce estimators that have higher variance. Although here we consider the simplest setting in which a single sample is used at each iteration j in Equation (5.14), the variance of the estimator can of course be reduced by using several samples per iteration.

5.3.3 Algorithm Based on LISSA

Provided that we use the LISSA procedure twice to construct two independent estimates \hat{w}_t, \hat{w}'_t of the optimal weight vector $W_{\Phi,t}^*$, it is straightforward to demonstrate that $\hat{g}_{\text{DL}}(s)$ (Equation (5.12)) becomes an unbiased estimate of the gradient of the loss $\mathcal{L}(\Phi)$ as $J \rightarrow \infty$; furthermore, for finite J its bias is controlled as a consequence from Lemma 13. We may then perform gradient descent with this estimate, adjusting the s^{th} row of the matrix Φ according to

$$\phi(s) \leftarrow \phi(s) - \alpha \hat{g}_{\text{DL}}(s), \quad (5.15)$$

where $\alpha \in [0, 1)$ is a suitable step size. Based on our derivation, we call this procedure the Danskin-LISSA algorithm. In practice, it is usually desirable to update ϕ for $N > 1$ rows at once and use $M > 1$ samples to estimate \hat{w}_i and \hat{w}'_i ; we give this more general form in Algorithm 1. Note that while larger values of J are desirable in order to reduce estimation bias, larger values of M and N contribute to reducing the variance of the gradient estimate \hat{g}_{DL} and speeding up the learning process.

An important case is when the row representation ϕ is given by a mapping that is parametrized by a collection of weights θ , in particular a neural network. In this case, Equation (5.15) should be replaced by an update rule that adjusts the weights θ . In practice, this can be done by determining the Jacobian $\frac{\partial\phi}{\partial\theta}$ of ϕ with respect to the weights θ , and applying the update

$$\theta \leftarrow \theta - \alpha \frac{\partial\phi}{\partial\theta} \hat{g}_{\text{DL}}(s).$$

An alternative particularly suited to automatic differentiation frameworks [Bradbury et al., 2018, Abadi et al., 2016, Paszke et al., 2019], is to define a loss function whose gradient corresponds to $\frac{\partial\phi}{\partial\theta} \hat{g}_{\text{DL}}(s)$. One can verify that the sample loss function

$$\begin{aligned} & \frac{1}{2} \left(\ell(\hat{w}_t + \hat{w}'_t) - \ell(\hat{w}_t) - \ell(\hat{w}'_t) \right) \\ \ell(w) &= \left(\phi(s)^\top \text{SG}(w) - \psi_t(s) \right)^2 \end{aligned}$$

satisfies this requirement, where SG denotes the stop-gradient operation (in the sense that $\nabla_{\theta} \text{SG}(w) = 0$). Additionally, the recursion in Equation (5.14) can be implemented efficiently by first computing the vector-matrix product $\phi(s_j)^\top \hat{\Delta}_{j-1}$ and then taking the outer product of the result with $\phi(s_j)$.

Algorithm 1: Danskin-LISSA

- 1: **Parameters:** Dimension $d \in \mathbb{N}^+$, $J, M, N \in \mathbb{N}^+$, $\alpha, \kappa_0 \in (0, 2)$
 - 2: **repeat**
 - 3: Sample independent rows $s_{1:N}, s'_{1:M}, s''_{1:M} \sim \xi$
 - 4: Sample a column $t \sim \lambda$
 - 5: $\hat{C} \leftarrow \text{LISSA}(\kappa_0, J)$
 - 6: $\hat{C}' \leftarrow \text{LISSA}(\kappa_0, J)$
 - 7: $w_t = \frac{\hat{C}}{M} \sum_{k=1}^M \phi(s'_k) \psi_t(s'_k)$
 - 8: $\hat{w}'_t = \frac{\hat{C}'}{M} \sum_{k=1}^M \phi(s''_k) \psi_t(s''_k)$
 - 9: $\hat{g}_{\text{LISSA}}(s_k) = \hat{w}'_t \left(\phi(s_k)^\top \hat{w}_t - \psi_t(s_k) \right)$
 - 10: $\phi(s_k) \leftarrow \phi(s_k) - \alpha \hat{g}_{\text{DL}}(s_k)$ for $k = 1, \dots, N$
 - 11: **until** satisfied
-

5.4 Related Work

LISSA Estimator. Agarwal et al. [2017] consider an empirical risk minimization problem with convex functions and rely on a Taylor expansion to estimate an Hessian Inverse. Our method contrasts with the paper from Agarwal et al. [2017] in several ways. Our algorithm consists in constructing an unbiased estimate of the gradient of a least square objective where the weights are expressed implicitly (non convex objective). To do so, we estimate the pseudoinverse of a covariance matrix by following an observation from Agarwal et al. [2017]. We write it as a Neumann series in a recursive formulation, and combine it with an unbiased estimate of the covariance from one row sample.

Streaming PCA. Oja [1982] and Krasulina [1970] proposed the original streaming PCA algorithms. They approximate the top eigenvector of a matrix through a stochastic approximation of the power method. Tang [2019] extends this method to other principal components but requires explicit normalization. Amid and Warmuth [2020] extends it without the need to explicitly performing orthonormalization after each gradient step at the cost of a batch having to be of size 1.

Pfau et al. [2019] recovers the subspace spanned by the top eigenfunctions of *symmetric* infinite dimensional matrices by parametrizing them with neural networks and performing gradient descent on a kernel-based loss. It is itself a generalization of slow feature analysis [Wiskott and Sejnowski, 2002] in the tabular setting. Deng et al. [2022] extends the objective from Gemp et al. [2021] to the function space and propose an algorithm to learn the top d -eigenfunctions of symmetric matrices by representing them with d neural networks. To find the principal subspace of a general infinite dimensional matrix Ψ , the approaches above require computing eigenfunctions of $\Psi\Psi^T$, which requires full row access to Ψ . By contrast, our method can recover the principal subspace of any infinite dimensional matrix using samples entries from rows of Ψ .

Low-rank Matrix Completion. In this setting, we observe a subset of entries from a data matrix and aim to find a low-rank matrix that matches these observations [Srebro and Jaakkola, 2003]. Matrix factorization is a common technique to solve this problem where the matrix of interest is expressed as a product ΦW . It can be solved efficiently by standard optimization algorithms [Sun and Luo, 2016]. Hardt [2014], Jain et al. [2013] rely on alternating minimization over the representation and weight matrices and guarantee convergence towards the true matrix. Other methods perform gradient descent [Li et al., 2019, Ye and Du, 2021] or stochastic gradient descent [Jin et al., 2016, Ge et al., 2015, De Sa et al., 2015]. Keshavan et al. [2010], Keshavan and Oh [2009] minimize simultaneously over the representation Φ and the weights W by gradient descent. Dai and Milenkovic [2010] first solves the inner optimization problem and find the optimal weight matrix W and then takes a gradient step on the outer optimization problem, with respect to the representation matrix Φ . The Grassmannian Rank-One Subspace Estimation (GROUSE) algorithm [Balzano et al., 2010] is a stochastic manifold gradient descent algorithm for tracking subspaces from incomplete data which was recently shown to be equivalent to Oja’s algorithm [Balzano, 2022]. In comparison, we consider the problem of learning low-dimensional embeddings of higher dimensional vectors through neural networks and propose an optimization procedure which performs gradient descent on the representation matrix Φ only and where the weight matrix W_{Φ}^* is expressed implicitly, as a function of Φ .

5.5 Experiments

We now conduct an empirical evaluation demonstrating that the Danskin-LISSA algorithm described in Section 5.3 recovers the d -dimensional principal subspace of different types of data: synthetic matrices, MNIST images [LeCun et al., 2010] and the successor measure for the modified PuddleWorld domain [Sutton, 1995]. In all cases, we measure convergence using the normalized subspace distance [Tang,

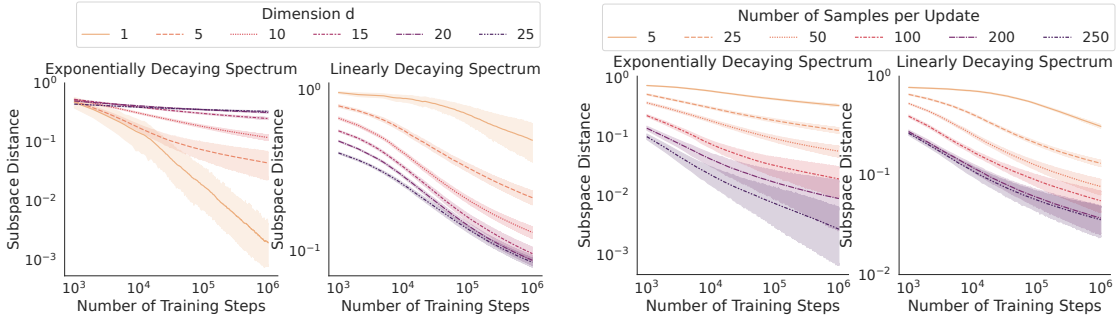


Figure 5.1: Subspace distance over the course of training LISSA for different dimensions (left, $L = 25, J = M = N = 5$) and for different total number of samples per update (right, $d = 10$) on synthetic matrices with a spectrum decaying linearly and exponentially, averaged over 30 seeds. Shaded areas represent estimates of 95% confidence intervals.

2019] between Φ and the principal subspace of Ψ :

$$1 - \frac{1}{d} \cdot \text{Tr} \left(F_d F_d^\top P_\Phi \right) \in [0, 1].$$

Here, F_d are the top- d left singular vectors of Ψ and $P_\Phi = (\Phi^\top \Phi)^\dagger \Phi^\top$ is the orthogonal projector onto the column space of Φ . For simplicity, we take $M = N = J$ in all experiments. The parameter $\kappa = \kappa_0 / \max_{s \in s_{1:J}} \|\phi(s)\|_2^2$, where κ_0 is a hyperparameter, is computed from the sampled feature vectors but we note that it can also be estimated online by a running average.

5.5.1 Synthetic Matrices

To begin, we consider a random matrix $\Psi \in \mathbb{R}^{50 \times 50}$ whose entries are sampled from a standard normal distribution. In order to study our algorithm’s behaviour under different conditions, we follow Gemp et al. [2021] and set the matrix’s singular values from 1000 to 1 linearly or exponentially (see Appendix 5.C.1 for more details). We selected the step size $\alpha = 0.001$ and the parameter $\kappa_0 = 1.9$ from a hyperparameter sweep. Figure 5.5 in Appendix 5.C.1 compares performance for different values of κ_0 , in particular illustrating how $\kappa_0 > 2$ fares poorly.

Figure 5.1, left illustrates that the Danskin-LISSA algorithm successfully recovers the d -dimensional principal subspace given sufficiently many training steps, with smaller values of d being easier to learn for the exponentially decaying spectrum (results for $d \geq 25$ are given in Appendix 5.C.1). However, we see that learning the

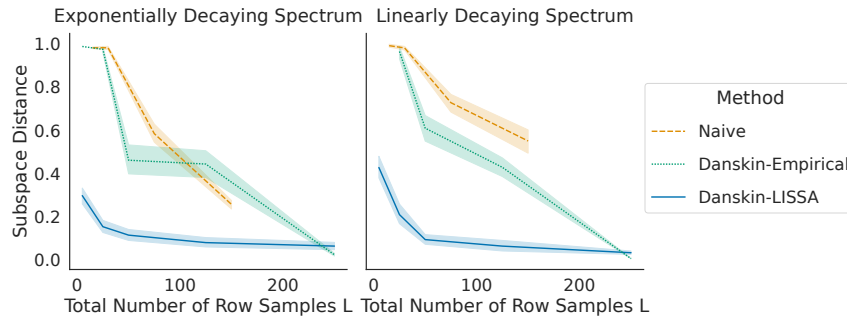


Figure 5.2: Subspace distance ($d = 10$) after 10^6 training steps according to the method used to estimate the loss gradient. Here, the x axis represents the total number of row samples L from the Φ matrix with $J = M = N$ ($L = 2J + 2M + N$ for the Danskin methods, $J + M + N$ for the naive method). Shaded areas represent estimates of 95% confidence intervals. Note that because we are sampling with replacement, the gradient estimate for $L = 250$ still differs from the gradient given in Lemma 12. (The naive method diverges for very small values of L).

subspace spanned by a representation of dimension $d = 25$ is easier than $d = 1$ for linearly decaying spectrum. Figure 5.1 right demonstrates that empirically, it is possible to obtain a reasonable approximation of the principal subspace even for a very smaller number of samples ($J = 1$ being the extreme), despite our theoretical expectation of a biased covariance estimate. As described in Section 5.3, the Danskin-LISSA approach stems from a combination of several algorithmic concepts. First, it uses two independent estimates of the weight vectors. Second, it embeds a LISSA procedure to estimate the inverse covariance matrix. To understand better their relative importance in the performance of the Danskin-LISSA algorithm, we compare it to two sample-based baselines which have access to the same amount of information and memory. The first one uses the naive gradient estimator described in Section 5.3. The second uses two separate weight estimates, following the derivation from Danskin’s theorem, but uses the inverse of the empirical covariance matrix rather than the LISSA procedure used in the Danskin-LISSA method – accordingly, we call this the Danskin-Empirical method. Figure 5.2 illustrates the bias-reducing advantage of the LISSA covariance estimator, in particular in the low-sample regime. The naive method, which constructs a single weight estimate, has high bias and underperforms compared to both of these methods.

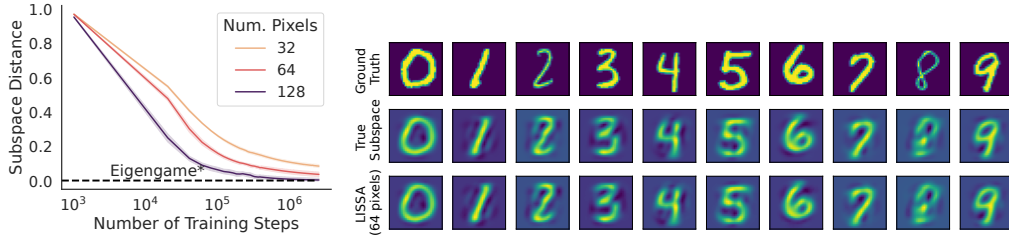


Figure 5.3: Training curves for LISSA on MNIST ($d = 16$) that updates only a subset of pixels at a time (left). *: see main text. Reconstruction on MNIST test images (right). First row show samples from test images. Second are images reconstructed from the true principal components of Ψ and third row are images reconstructed from the principal components learnt by Danskin-LISSA ($N = 64$). Reconstruction MSE errors for true components and Danskin-LISSA are 21.46 and 21.53 respectively.

5.5.2 MNIST Dataset

We now consider learning the principal subspace of MNIST images from a training dataset with the Danskin-LISSA algorithm. We represent the data as a matrix $\Psi \in \mathbb{R}^{784 \times 60000}$ where each column is a 28×28 sample image (flattened to size 784) of one of the ten possible digits and from which the mean image has been subtracted. To accelerate learning speed we use the second-order Adam optimizer [Kingma and Ba, 2015]. Figure 5.3 shows that it is possible to effectively learn the principal subspace of this data even while updating as few as 32 pixels (rows) at a time; naturally, using more samples per step results in improved learning speed. As a point of comparison, we provide the subspace distance obtained by Eigengame [Gemp et al., 2021], a state-of-the-art method that performs PCA by sampling full columns (images) at a time.

To quantify the goodness of the representation learnt on the MNIST training set, we use it to reconstruct MNIST images on the test set. Denoting $\Psi_{\text{test}} \in \mathbb{R}^{784 \times 10000}$ the test dataset and $\Phi \in \mathbb{R}^{784 \times d}$ a representation learnt from the training set, the reconstructed images on the test set are given by $P_{\Phi} \Psi_{\text{test}}$ where P_{Φ} denotes the orthogonal projector onto the column space of Φ . Figure 5.3, right, shows that the MNIST digits reconstructed from the subspace learnt by Danskin-LISSA qualitatively look similar to the images reconstructed from the true principal components of the training set and achieve a similar reconstruction error.

5.5.3 Learning the Successor Measure

In reinforcement learning (RL), the successor representation [Dayan, 1993] encodes an agent’s future trajectories from any given state in terms of the visitation frequency to various states. Of immediate relevance, it is often used as a building block in representation learning for RL, in particular by directly learning its principal subspace [Mahadevan and Maggioni, 2007, Behzadian and Petrik, 2018, Machado et al., 2018]. Its extension to continuous state spaces is called the successor measure [Blier et al., 2021], and is naturally described by an infinite dimensional matrix. Our last experiment illustrates how the Danskin-LISSA algorithm can be used to approximate the principal subspace of the successor measure of the Puddle World domain [Sutton, 1995].

In our version of this environment, traversing puddles requires more time, resulting in asymmetric successor measure; details of the environment and the reinforcement learning framework are given in Appendix 5.C.3. Here, $s \in [0, 1]^2$ corresponds to a particular two-dimensional state in the environment. For a collection of sets $\mathcal{X} = \{X \subset [0, 1]^2\}$ to be described below, we define the successor measure as

$$\Psi(s, X) = \sum_{t \geq 0} \gamma^t \mathbb{P}(S_t \in X \mid S_0 = s), \quad \gamma \in (0, 1)$$

The successor measure describes the expected, discounted number of visits to the set X when the agent begins in state s and moves randomly. We take $\gamma = 0.99$. Compared to the experiments of the previous sections, we parametrize the representation by a neural network. We are interested in understanding the degree to which this neural network can be trained to approximate the d -dimensional principal subspace of the successor measure. We take the collection X to be the set of non-overlapping cells of a 100×100 grid (illustrated by Figure 5.4). For computational reasons, we assign the same value of $\Psi(\cdot, X)$ to all states within a grid cell; this value is computed by 1,000 truncated Monte-Carlo rollouts from a start state sampled uniformly at random within a cell. This produces a $10,000 \times 10,000$ matrix which we treat as ground truth for measuring the accuracy of our predicted subspace.

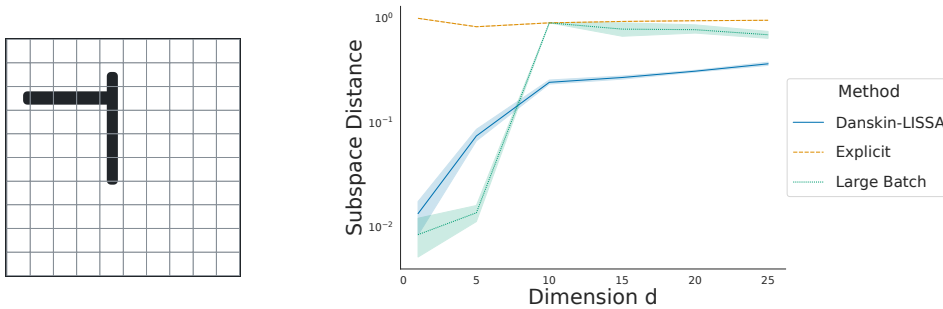


Figure 5.4: The Puddle World domain [Sutton, 1995], with the shaded area indicating regions where the agent moves slowly (left). In our experiments, each grid cell is associated with a column of the implied data matrix. Subspace distance as a function of the dimension d after 10^8 gradient steps for three methods: Danskin-LISSA, Explicit, and the Large Batch baseline (right).

To gain an understanding of the effectiveness of our method, we compare it with two other gradient-based methods commonly used in reinforcement learning. As the name indicates, the Explicit method maintains a weight vector w_i for each column and relies on the pair of updates from Equation (5.7), similar to the method used by Bellemare et al. [2019], Lyle et al. [2021]. Note that we present this method only for completeness, as it is not applicable to an infinite number of columns and may otherwise carry an impractically large memory cost. The Large Batch method, on the other hand, estimates the weight vector w_i using ϕ and Ψ evaluated at center of each of the 10,000 grid cells (close in spirit to the Naïve method of Subsection 5.5.1).

All three methods use Adam [Kingma and Ba, 2015] to optimize a two-layer MLP with 512 hidden units and ReLU activations. We take $J = M = N = 50$ for Danskin-LISSA and $N = 250$ for the two other methods. The step size α was tuned for each method according to a small hyperparameter sweep and after 10^8 gradient steps averaged across 5 runs. Details outlining these sweeps and complete experimental methodology can be found in Appendix 5.C.

Figure 5.4, right compares the final subspace distance of these three algorithms for various values of d . We find that the performance of the Danskin-LISSA algorithm degrades gracefully as d is increased, while the Large Batch method is only practical for small values of d . In part, this is explained by the fact that even with such a large batch, there is a residual bias in the latter method’s covariance

estimate. The poor performance of the Explicit method is explained by the fact that a single column is updated at any given time, resulting in stale weight vectors w_i . Although in practice this can be mitigated by updating multiple columns at once, the result illustrates an important pitfall with the use of an explicit weight vector.

5.6 Discussion & Conclusion

In this paper, we presented an algorithm that learns principal components of very large or infinite dimensional matrices by stochastic gradient descent. Our experiments on synthetic matrices and MNIST images demonstrate that indeed the method converges towards their top principal subspace. Our analysis on the Puddle World domain also demonstrates that our algorithm can learn a low-dimensional, neural-network state representation. This algorithm would benefit online PCA applications where the columns, whose total number can be unknown, and the rows of the data matrix of interest are sampled i.i.d. at each time step. For instance, in deep reinforcement learning (RL), training a network on supervised auxiliary predictions results in its representation corresponding to the principal components of this set of tasks, assuming the network is other unconstrained [Bellemare et al., 2019]. The rows of the auxiliary task matrix are the states and the columns are value functions, for instance corresponding to different discount factors sampled from the interval $(0, 1)$ [Fedus et al., 2019]. Incorporating the Danskin-LISSA procedure within a deep RL architecture may provide performance improvements by incorporating more knowledge about the world into the network’s representation.

For simplicity, in this paper we assumed that all samples used in computing a given gradient estimate are drawn independently. In practice, samples are naturally expensive and it may appear undesirable to require a total of $N + 2J + 2M$ for a single gradient estimate. However, one can improve on this state of affairs by permuting the order in which samples from the batch are presented, constructing different gradient estimates from these permutations, and noting that the average of multiple unbiased estimates remains unbiased (and generally has lower variance).

Acknowledgements

The authors would like to thank Ian Gemp, Mathieu Blondel, Matthieu Geist and the anonymous reviewers for useful discussions and feedback on this paper.

We also thank the Python community [Van Rossum and Drake Jr, 1995, Oliphant, 2007] for developing tools that enabled this work, including NumPy [Oliphant, 2006, Walt et al., 2011, Harris et al., 2020], SciPy [Jones et al., 2001], Matplotlib [Hunter, 2007] and JAX [Bradbury et al., 2018].

5.A Proofs for Section 5.2

Lemma 14. *Following the notations from Section 5.2, we have*

$$W_{\Phi}^* = (\Phi^{\top} \Xi \Phi)^{\dagger} \Phi^{\top} \Xi \Psi \quad (5.16)$$

Proof. For a fixed $\Phi \in \mathbb{R}^{S \times d}$,

$$\begin{aligned} \nabla_W \mathcal{L}(\Phi, W) &= \nabla_W \|\Xi^{1/2}(\Phi W - \Psi)\Lambda^{1/2}\|_F^2 \\ &= 2\Phi^{\top} \Xi (\Phi W - \Psi) \Lambda \end{aligned}$$

$$\begin{aligned} \nabla_W \mathcal{L}(\Phi, W) = 0 &\iff 2\Phi^{\top} \Xi (\Phi W_{\Phi}^* - \Psi) \Lambda = 0 \\ &\iff \Phi^{\top} \Xi (\Phi W_{\Phi}^* - \Psi) = 0 \text{ as } \lambda(t) > 0 \text{ for all } t \in \mathcal{T} \\ &\iff \Phi^{\top} \Xi \Phi W_{\Phi}^* = \Phi^{\top} \Xi \Psi \\ &\iff W_{\Phi}^* = (\Phi^{\top} \Xi \Phi)^{\dagger} \Phi^{\top} \Xi \Psi \end{aligned}$$

□

Proposition 1. *Let $GL_d(\mathbb{R})$ be the set of $d \times d$ invertible matrices. Assume Ψ has strictly decreasing singular values and $\text{rank}(\Psi) = r < \infty$. Write $\Psi = F \Sigma B^{\top}$ for the SVD of Ψ with respect to the inner products $\langle x, y \rangle_{\Xi}$ for all $x, y \in \mathbb{R}^S$ and $\langle x, y \rangle_{\Lambda}$ for all $x, y \in \mathbb{R}^T$. For an integer $\ell \in \{1, \dots, S\}$, let $F_{\ell} \in \mathbb{R}^{S \times \ell}$ be the matrix*

containing the first ℓ columns of F (sorted by decreasing singular value). For a fixed $d \in \{1, \dots, r\}$,

$$\arg \min_{\Phi \in \mathbb{R}^{S \times d}} \min_{W \in \mathbb{R}^{d \times T}} \|\Xi^{\frac{1}{2}}(\Phi W - \Psi)\Lambda^{\frac{1}{2}}\|_F^2 = \{\Phi \in \mathbb{R}^{S \times d} \mid \exists M \in GL_d(\mathbb{R}), \Phi = F_d M\}. \quad (5.17)$$

Proof. We have $\Psi = F\Sigma B^\top$ where $F \in \mathbb{R}^{S \times r}$, $\Sigma \in \mathbb{R}^{r \times r}$ and $B \in \mathbb{R}^{T \times r}$ satisfy $F^\top \Xi F = I, B^\top \Lambda B = I$. Let F_d, Σ_d and B_d the matrices containing the first d columns of F, Σ and B respectively. For a fixed $\Phi \in \mathbb{R}^{S \times d}$ and if Φ is full rank, the unique solution of $\min_{W \in \mathbb{R}^{d \times T}} \|\Xi^{\frac{1}{2}}(\Phi W - \Psi)\Lambda^{\frac{1}{2}}\|_F^2$ is given by $W_\Phi^* = (\Phi^\top \Xi \Phi)^{-1} \Phi^\top \Xi \Psi$. When Φ is orthonormal with respect to the inner product induced by Ξ , we have $\Phi^\top \Xi \Phi = I$ and $W_\Phi^* = \Phi^\top \Xi \Psi$. Moreover, $\text{rank}(\Phi W_\Phi^*) \leq \min(\text{rank}(\Phi), \text{rank}(\Phi^\top \Xi \Psi)) \leq \min(d, \min(d, S, r)) = d$. By the Eckart-Young theorem, given a target matrix Ψ , the best approximating matrix of rank at most d , with respect to the norm induced by Ξ , is $F_d \Sigma_d B_d^\top$ which can be written in terms of an orthogonal projection as follows $F_d F_d^\top \Xi \Psi$. By identification, $\Phi W_\Phi^* = \Phi(\Phi^\top \Xi \Psi) = F_d(F_d^\top \Xi \Psi)$ and $\Phi = F_d$ is a solution to Equation (5.17).

As we can turn the basis Φ for $\text{span}(F_d)$ into any other basis $\Phi' = \Phi R$ with $R \in \mathbb{R}^{d \times d}$ an invertible matrix, the set of solutions for Φ is $\{F_d R : R \in \mathbb{R}^{d \times d} \text{ invertible}\}$

□

5.B Proofs for Section 5.3

Let $\Xi = \mathbb{E}_{s \sim \nu}[e_s e_s^\top]$ and $\Lambda = \mathbb{E}_{t \sim \Lambda}[e_t e_t^\top]$.

Lemma 15. *The j -LISSA estimator $\widehat{\Delta}_j$ is an unbiased estimator of the partial Neumann series defined in Equation (5.13). That is, given j samples $s_{1:j} = \{s_1, s_2, \dots, s_j\}$ drawn i.i.d. from ξ , we have that*

$$\mathbb{E}_{s_{1:j} \sim \xi} [\widehat{\Delta}_j] = \kappa \sum_{i=0}^j (I - \kappa \Phi^\top \Xi \Phi)^i$$

Proof. By induction.

$$\begin{aligned}\mathbb{E}[\widehat{\Delta}_0] &= \mathbb{E}[\kappa I] = \kappa I \text{ and } \kappa \sum_{i=0}^0 (I - \kappa \Phi^\top \Xi \Phi)^i = \kappa I \\ \mathbb{E}_{s_1 \sim \xi}[\widehat{\Delta}_1] &= \mathbb{E}_{s_1 \sim \xi}[\kappa I + (I - \kappa \phi_{s_1} \phi_{s_1}^\top) \kappa I] = \kappa I + \kappa (I - \kappa \Phi^\top \Xi \Phi) \\ \text{as } \mathbb{E}[\phi_i \phi_i^\top] &= \Phi^\top \Xi \Phi \text{ and } \kappa \sum_{i=0}^1 (I - \kappa \Phi^\top \Xi \Phi)^i = \kappa I + \kappa (I - \kappa \Phi^\top \Xi \Phi).\end{aligned}$$

Let's suppose that $\mathbb{E}_{s_{1:j-1} \sim \xi}[\widehat{\Delta}_{j-1}] = \kappa \sum_{i=0}^{j-1} (I - \kappa \Phi^\top N \Phi)^i$. Then,

$$\begin{aligned}\mathbb{E}_{s_{1:j} \sim \xi}[\widehat{\Delta}_j] &= \mathbb{E}_{s_{1:j}}[\kappa I + (I - \kappa \phi_{s_j} \phi_{s_j}^\top) \widehat{\Delta}_{j-1}] \\ &= \kappa I + \mathbb{E}_{s_{1:j}}[(I - \kappa \phi_{s_j} \phi_{s_j}^\top) \widehat{\Delta}_{j-1}] \\ &= \kappa I + \mathbb{E}_{s_j \sim \xi}[I - \kappa \phi_{s_j} \phi_{s_j}^\top] \mathbb{E}_{s_{1:j-1} \sim \nu}[\widehat{\Delta}_{j-1}] \\ &= \kappa I + (I - \kappa \Phi^\top N \Phi) \kappa \sum_{i=0}^{j-1} (I - \kappa \Phi^\top N \Phi)^i \\ &= \kappa \sum_{i=0}^j (I - \kappa \Phi^\top N \Phi)^i\end{aligned}$$

Hence, the conclusion. \square

Lemma 13 (Bias of LISSA). *For $0 < \kappa < \sup_{s_{1:j}} 2\|\phi(s_i)\|_2^{-2}$, the bias of $\widehat{\Delta}_j$ with respect to $(\Phi^\top \Xi \Phi)^\dagger$ is given by*

$$\mathbb{E}(\widehat{\Delta}_j) - (\Phi^\top \Xi \Phi)^\dagger = -(\Phi^\top \Xi \Phi)^\dagger (I - \kappa \Phi^\top \Xi \Phi)^{j+1}$$

In particular, this bias asymptotically vanishes, in the sense that

$$\lim_{j \rightarrow \infty} \mathbb{E}(\widehat{\Delta}_j) - (\Phi^\top \Xi \Phi)^\dagger = 0.$$

Proof.

$$\begin{aligned}\text{bias}(\widehat{\Delta}_j) &= \mathbb{E}(\widehat{\Delta}_j) - (\Phi^\top \Xi \Phi)^\dagger \\ &= \kappa \sum_{i=0}^j (I - \kappa \Phi^\top \Xi \Phi)^i - (\Phi^\top \Xi \Phi)^\dagger \text{ by Lemma 15} \\ &= \kappa (I - (I - \kappa \Phi^\top \Xi \Phi))^\dagger (I - (I - \kappa \Phi^\top \Xi \Phi)^{j+1}) - (\Phi^\top \Xi \Phi)^\dagger \\ &= -(\Phi^\top \Xi \Phi)^\dagger (I - \kappa \Phi^\top \Xi \Phi)^{j+1}\end{aligned}$$

In the third line, we use the closed form of a geometric series. \square

Theorem 8. Let $e_s \in \mathbb{R}^S$ denote a basis vector. Given two independent unbiased estimates \hat{C} and \hat{C}' of the inverse covariance, for $s \sim \xi$, the gradient estimate $\hat{g}_{\text{DL}}(s)$ given in Equation (5.12) satisfies

$$\mathbb{E}[e_s \hat{g}_{\text{DL}}(s)^\top] = \Xi(\Phi W_\Phi^* - \Psi)\Lambda W_\Phi^{*\top}.$$

Proof. By definition,

$$\hat{g}_{\text{DL}}(s) = \hat{w}'_i \left(\phi(s)^\top \hat{w}_i - \psi_i(s) \right)$$

Plugging in $\hat{w}_i = \hat{C}\phi(s')\psi_i(s')$ and $\hat{w}'_i = \hat{C}'\phi(s'')\psi_i(s'')$, we have

$$\hat{g}_{\text{DL}}(s)^\top = \left(\phi(s)^\top \hat{C}\phi(s')\psi_i(s') - \psi_i(s) \right) \left(\hat{C}'\phi(s'')\psi_i(s'') \right)^\top$$

Now taking the expectation,

$$\begin{aligned} & \mathbb{E}_{s,s',s'',s_{1:n},s'_{1:n},i} [e_s \hat{g}_{\text{DL}}(s)^\top] \\ &= \mathbb{E}_{s,s',s'',s_{1:n},s'_{1:n},i} \left[e_s \left(\phi(s)^\top \hat{C}\phi(s')\psi_i(s') - \psi_i(s) \right) \left(\hat{C}'\phi(s'')\psi_i(s'') \right)^\top \right] \\ &= \mathbb{E}_{s,i} \left[e_s \left(\phi(s)^\top \mathbb{E}_{s_{1:n}} [\hat{C}] \mathbb{E}_{s'} [\phi(s')\psi_i(s')] - \psi_i(s) \right) \left(\mathbb{E}_{s'_{1:n}} [\hat{C}'] \mathbb{E}_{s''} [\phi(s'')\psi_i(s'')] \right)^\top \right] \\ &= \mathbb{E}_{s,i} \left[e_s \left(e_s^\top \Phi \mathbb{E}_{s_{1:n}} [\hat{C}] \mathbb{E}_{s'} [\Phi^\top e_{s'} e_{s'}^\top \Psi e_i] - e_s^\top \Psi e_i \right) \left(\mathbb{E}_{s'_{1:n}} [\hat{C}'] \mathbb{E}_{s''} [\Phi^\top e_{s''} e_{s''}^\top \Psi e_i] \right)^\top \right] \\ &= \mathbb{E}_{s,i} \left[e_s e_s^\top \left(\Phi \mathbb{E}_{s_{1:n}} [\hat{C}] \mathbb{E}_{s'} [\Phi^\top e_{s'} e_{s'}^\top \Psi] - \Psi \right) e_i e_i^\top \mathbb{E}_{s''} [\Psi^\top e_{s''} e_{s''}^\top \Phi] \left(\mathbb{E}_{s'_{1:n}} [\hat{C}'] \right)^\top \right] \\ &= \mathbb{E}_s e_s e_s^\top \left(\Phi \mathbb{E}_{s_{1:n}} [\hat{C}] \Phi^\top \mathbb{E}_{s'} [e_{s'} e_{s'}^\top] \Psi - \Psi \right) \mathbb{E}_i [e_i e_i^\top] \Psi^\top \mathbb{E}_{s''} [e_{s''} e_{s''}^\top] \Phi \mathbb{E}_{s'_{1:n}} [\hat{C}'] \\ &= \Xi \left(\Phi \mathbb{E}_{s_{1:n}} [\hat{C}] \Phi^\top \Xi \Psi - \Psi \right) \Lambda \left(\Psi^\top \Xi \Phi \mathbb{E}_{s'_{1:n}} [\hat{C}'] \right)^\top \end{aligned}$$

where in the last line, we used the fact that $\Xi = \mathbb{E}_{s \sim \nu} [e_s e_s^\top]$ and $\Lambda = \mathbb{E}_{t \sim \Lambda} [e_t e_t^\top]$.

Now, given two unbiased estimators \hat{C} and \hat{C}' , we have

$$\mathbb{E}_{s_{1:n}} [\hat{C}] = (\Phi \Xi \Phi^\top)^\dagger \text{ and } \mathbb{E}_{s'_{1:n}} [\hat{C}'] = (\Phi \Xi \Phi^\top)^\dagger$$

It then follows that

$$\begin{aligned} \mathbb{E}_{s,s',s'',s_{1:n},s'_{1:n},i} [e_s \hat{g}_{\text{DL}}^\top(s)] &= \Xi \left(\Phi (\Phi \Xi \Phi^\top)^\dagger \Phi^\top \Xi \Psi - \Psi \right) \Lambda \left(\Psi^\top \Xi \right) \Phi (\Phi \Xi \Phi^\top)^\dagger \\ &= \Xi \left(\Phi (\Phi \Xi \Phi^\top)^\dagger \Phi^\top \Xi \Psi - \Psi \right) \Lambda \left((\Phi \Xi \Phi^\top)^\dagger \Phi^\top \Xi \Psi \right)^\top \\ &= \Xi (\Phi W_\Phi^* - \Psi) \Lambda (W_\Phi^*)^\top \\ &\propto \nabla_\Phi \mathcal{L}(\Phi) \end{aligned}$$

□

5.C Additional Experimental Results

5.C.1 Synthetic Matrices

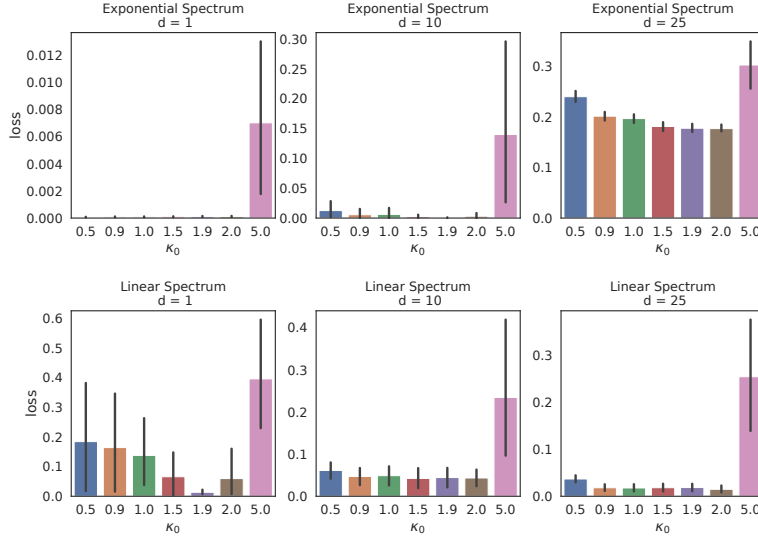


Figure 5.5: Subspace distance after 10^6 training steps of the LISSA algorithm for different κ_0

We follow the experimental protocol from Gemp et al. [2021]. We initialize $\Psi \in \mathbb{R}^{50 \times 50}$ randomly from a normal distribution. We compute its SVD such that $\Psi = F\Sigma B$. Let $\Sigma_{\text{linear}} = \text{diag}(1, \dots, 1000)$ and $\Sigma_{\text{exp}} = \text{diag}(10^0, \dots, 10^3)$. We rescale the matrix Ψ such that $\Psi_{\text{linear}} = F\Sigma_{\text{linear}}B$ and $\Psi_{\text{exp}} = F\Sigma_{\text{exp}}B$. The matrix $\Phi \in \mathbb{R}^{S \times d}$ is also initialized randomly from a standard normal distribution. We swept over the step size α and chose $\alpha = 0.001$ which was working well in all the synthetic experiments. We used the SGD optimizer but found that there was not a big performance difference with the Adam optimizer [Kingma and Ba, 2015] in most of these synthetic experiments. In Figure 5.5, we also swept over the hyperparameter κ_0 and found that $\kappa_0 = 1.9$ was performing well across dimensions and for both linear and exponential spectra. We trained the Danskin-LISSA method for 10^6 time steps. As a complement to Figure 5.1, we show in Figure 5.6 the training curves of the Danskin-LISSA algorithm for a broader range of dimensions d . For the exponential spectrum, when $d \geq 25$, larger dimensions are easier to learn. This is

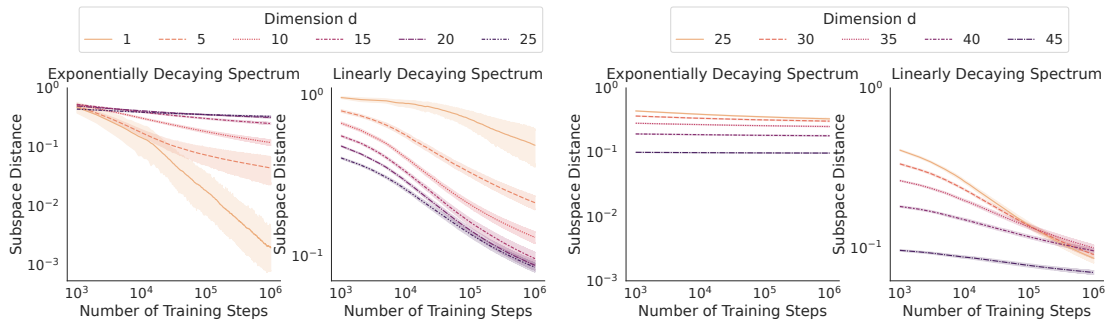


Figure 5.6: Subspace distance over the course of training LISSA for different dimensions on synthetic matrices with a spectrum decaying linearly and exponentially, averaged over 30 seeds. The total number of samples used is 50. Shaded areas represent 95% confidence intervals.

the opposite trend to the behavior found when $d \leq 25$ where smaller dimensions are easier to learn. For the linearly decaying spectrum, when $d \geq 25$, larger dimensions are easier to learn which is also the same trend as what we observed for $d \leq 25$.

5.C.2 MNIST

We found that the Adam optimizer [Kingma and Ba, 2015] performed best for our MNIST experiments. We performed a sweep over the step-size α and found that $\alpha = 0.005$ worked best for 128 and 64 pixels. $\alpha = 0.01$ performed best for 32 pixels. We trained the Danskin-LISSA algorithm for 2.5×10^6 steps.

5.C.3 Puddle World

A Puddle World [Sutton, 1995] is a square arena, with x, y both in $[0, 1]$. It has a continuous state space and a discrete action space. There are four actions (up, down, left, right) that move the agent by 0.05 in each of the corresponding directions. A random gaussian noise with standard deviation 0.01 is also added to transitions in both directions. For our experiments, we used the same puddle configuration found in [Sutton, 1995]. This configuration contains two puddles. The first puddle lies between the points $(0.1, 0.75)$ and $(0.45, 0.75)$ with a radius of 0.1. The second puddle lies between the points $(0.45, 0.4)$ and $(0.45, 0.8)$, also with a radius of 0.1. While the original Puddle World gives negative rewards for being in a puddle, our puddles instead cause a slowing affect by a factor of 0.5. That is, when in a puddle,

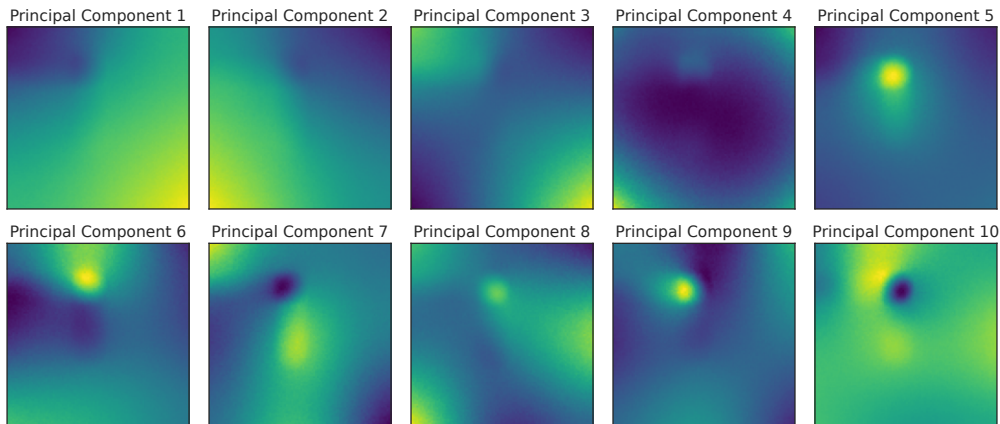


Figure 5.7: First 10 principal components of the successor measure of the Puddle World domain.

the agent only moves by 0.025 in each direction. The puddles compound, meaning that in the area where the two puddles overlap the agent will only move a distance of 0.0125. We chose to use slowing puddles because our task is reward-agnostic, and the successor measure task that we chose would capture the dynamics of the slowing puddles. We visualize in Figure 5.7 the top-10 principal components of the successor measure of Puddle World, demonstrating that they are non-trivial. The successor measure was computed using 1000 Monte Carlo rollouts from each starting grid cell, truncated after 700 steps. We used a discount factor $\gamma = 0.99$. We subtracted the row sums to center-mean each column of the ground truth matrix $\Psi \in \mathbb{R}^{10^4 \times 10^4}$.

For each of the methods, we performed a sweep of learning rates and optimizers (between Adam and SGD) and found that Adam with a learning rate of 10^{-4} worked well across the board. We ran each method for 100 million gradient steps. For Danskin-LISSA, we kept κ fixed at 1.9, which we found worked well in our previous experiments. Danskin-LISSA used a batch size of 50 for each of its 5 batches, while Large Batch and Explicit used a main batch size of 250 to ensure that each method saw the same number of samples. To compute $\phi(s)$ we used a two hidden-layer MLP with 512 hidden units per layer.

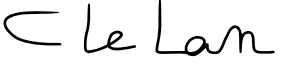
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor (**only required where there isn't already a statement of contribution within the paper itself**).


Title of Paper	A Novel Stochastic Gradient Descent Algorithm for Learning Principal Subspaces
Publication Status	<input checked="" type="checkbox"/> Published <input type="checkbox"/> Accepted for Publication <input type="checkbox"/> Submitted for Publication <input type="checkbox"/> Unpublished and unsubmitted work written in a manuscript style
Publication Details	Charline Le Lan, Joshua Greaves, Jesse Farebrother, Mark Rowland, Fabian Pedregosa, Rishabh Agarwal, Marc G. Bellemare. A Novel Stochastic Gradient Descent Algorithm for Learning Principal Subspaces. <i>In International Conference on Artificial Intelligence and Statistics (AISTATS) 2023</i>

Student Confirmation

Student Name:	Charline Le Lan		
Contribution to the Paper	I led the project, developed the methodology, proved most theoretical results, wrote the first draft of the paper, implemented the first version of the code used for the synthetic experiments. Josh ran and created the plots for the synthetic experiments, implemented the puddle world codebase. Jesse implemented the implicit method with function approximation and ran some synthetic and puddle world experiments. Mark proofread the methodology of the paper giving comments on theoretical aspects of the writing, Fabian helped with the theoretical applicability of the LISSA estimator to our setting, Rishabh ran and made plots for the MNIST experiment, Marc suggested the initial project idea, advised the project, provided feedback / edits on the manuscript.		
Signature		Date	March 23, 2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Marc G. Bellemare			
Supervisor comments	This was an incredibly challenging piece of work to execute, because of subtle implementation details in the algorithm.		
Signature		Date	13/04/23

This completed form should be included in the thesis, at the end of the relevant chapter.

6

Bootstrapped Representations in Reinforcement Learning

Abstract

In reinforcement learning (RL), state representations are key to dealing with large or continuous state spaces. While one of the promises of deep learning algorithms is to automatically construct features well-tuned for the task they try to solve, such a representation might not emerge from end-to-end training of deep RL agents. To mitigate this issue, auxiliary objectives are often incorporated into the learning process and help shape the learnt state representation. Bootstrapping methods are today’s method of choice to make these additional predictions. Yet, it is unclear which features these algorithms capture and how they relate to those from other auxiliary-task-based approaches. In this paper, we address this gap and provide a theoretical characterization of the state representation learnt by temporal difference learning [Sutton, 1988]. Surprisingly, we find that this representation differs from the features learned by Monte Carlo and residual gradient algorithms for most transition structures of the environment in the policy evaluation setting. We describe the efficacy of these representations for policy evaluation, and use our theoretical analysis to design new auxiliary learning rules. We complement our theoretical results with an empirical comparison of these learning rules for different cumulant functions on classic domains such as the Four Rooms domain [Sutton et al., 1999] and Mountain Car [Moore, 1990] and demonstrate that these pretrained representations speed up online learning.

6.1 Introduction

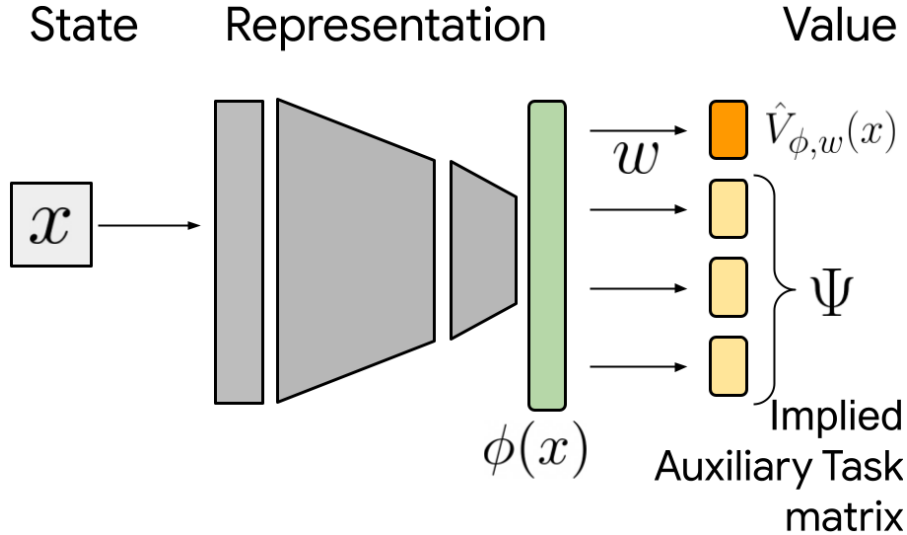


Figure 6.1: In deep RL, we see the penultimate layer of the network as the representation ϕ which is linearly transformed into a value prediction $\hat{V}_{\phi,w}$ and auxiliary predictions $\Psi(x)$ by bootstrapping methods.

The process of representation learning is crucial to the success of reinforcement learning at scale. In deep reinforcement learning, a neural network is used to parameterise a representation ϕ which is linearly mapped into a *value function* [Figure 6.1; Yu and Bertsekas, 2009, Bellemare et al., 2019, Levine et al., 2017]; this approach often leads to state-of-the-art performance in the field [Mnih et al., 2015]. State representations are key to the stability and quality of this learning process.

However, a representation supporting the downstream task of interest might not emerge from end-to-end training. Hence, auxiliary objectives are often incorporated into the training process to help the agent combine its inputs into useful features [Sutton et al., 2011, Jaderberg et al., 2017, Bellemare et al., 2017, Lyle et al., 2021] and the resulting network’s representation can help the agent estimate the value function. To construct representations supporting these characteristics, different kind of auxiliary tasks have thus been incorporated into the learning process such as controlling visual aspects of observed states [Jaderberg et al., 2017], predicting the values of several policies [Bellemare et al., 2019, Dabney et al., 2021], predicting values over multiple discount factors [Fedus et al., 2019] or prediction of next state

observations [Jaderberg et al., 2017, Gelada et al., 2019] and rewards [Dabney et al., 2021, Lyle et al., 2021, Farebrother et al., 2023].

While these tasks have mainly been trained by bootstrapping, a precise characterization of the resulting representation is lacking. This paper aims to fill this gap. We study the representations learnt by TD learning when training auxiliary tasks consisting in predicting the expected return of a fixed policy for several cumulant functions (Section 6.3). More generally, this analysis informs bootstrapped representations arising from algorithms such as Q-learning [Watkins and Dayan, 1992], n-step Q-learning [Hessel et al., 2018, Kapturowski et al., 2019, Schwarzer et al., 2021] or Retrace [Munos et al., 2016]. Our key insight is that the way we train these value functions, for instance by TD learning, Monte Carlo or residual gradient, influences the resulting features. In particular, we show that when trained by TD learning, these features converge to the top- d real invariant subspace of the transition matrix P^π , when it exists (Theorem 9). We present an empirical evaluation that supports our theoretical characterizations and show the importance of the choice of a learning rule to learn the value function in Section 6.5.

In Section 6.4, we characterise the goodness of these representations by quantifying the approximation error of a linear prediction of the value function from these frozen representations in the TD learning and batch Monte Carlo settings (Subsection 6.4.1). We find that to minimize this error, the cumulants need to depend on the dynamics of the environment but in a different way whether we learn the main value function by batch Monte Carlo or TD learning. Then, we show random cumulants which have been used in the literature [Lyle et al., 2021, Farebrother et al., 2023] can be good pseudo-reward functions for some particular structures of the successor representation [Dayan, 1993] by providing an error bound that arises from sampling a small number of random pseudo-reward functions (Subsection 6.4.2).

Finally, we find that one way to improve this bound is to sample pseudo-reward functions which depend on the dynamics of the environment and inspired by this observation, we propose a novel auxiliary task method with adaptive cumulants

and show that the resulting pretrained features outperform training from scratch on the Four Rooms and Mountain Car domains Subsection 6.5.3.

6.2 Background

We consider a Markov decision process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$ [Puterman, 1994] with finite state space \mathcal{S} , finite set of actions \mathcal{A} , transition kernel $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{P}(\mathcal{S})$, deterministic reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow [-R_{\max}, R_{\max}]$, and discount factor $\gamma \in [0, 1)$. A stationary policy $\pi : \mathcal{S} \rightarrow \mathcal{P}(\mathcal{A})$ is a mapping from states to distributions over actions, describing a particular way of interacting with the environment. We denote the set of all policies by Π . We write $P_\pi : \mathcal{S} \rightarrow \mathcal{P}(\mathcal{S})$ the transition kernel induced by a policy $\pi \in \Pi$

$$P_\pi(s, s') = \sum_{a \in \mathcal{A}} \mathcal{P}(s, a)(s') \pi(a | s)$$

and $r_\pi : \mathcal{S} \rightarrow [-R_{\max}, R_{\max}]$ the expected reward function

$$r_\pi(s) = \mathbb{E}_\pi[\mathcal{R}(S_0, A_0) | S_0 = s, A_0 \sim \pi(\cdot | S_0)].$$

For any policy $\pi \in \Pi$, the value function $V^\pi(s)$ measures the expected discounted sum of rewards received when starting from state $s \in \mathcal{S}$ and acting according to π :

$$V^\pi(s) := \mathbb{E}_{\pi, \mathcal{P}} \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(S_t, A_t) | S_0 = s, A_t \sim \pi(\cdot | S_t) \right].$$

It satisfies the Bellman equation [Bellman, 1957]

$$V^\pi(s) = r_\pi(s) + \gamma \mathbb{E}_{S' \sim P_\pi(\cdot | s)} [V^\pi(S')],$$

which can be expressed using vector notation [Puterman, 1994] (viewing r_π and V^π as vectors in $\mathbb{R}^{\mathcal{S}}$ and P_π as an $\mathbb{R}^{\mathcal{S} \times \mathcal{S}}$ transition matrix) as

$$V^\pi = r_\pi + \gamma P_\pi V^\pi = (I - \gamma P_\pi)^{-1} r_\pi.$$

We are interested in approximating the value function V^π using a linear combination of features [Yu and Bertsekas, 2009, Levine et al., 2017, Bellemare et al., 2019]. We call the mapping $\phi : \mathcal{S} \rightarrow \mathbb{R}^d$ a *state representation*, where $d \in \mathbb{N}^+$. Usually, we are

interested in reducing the number of parameters needed to approximate the value function and have $d \ll |\mathcal{S}|$. Given a feature vector $\phi(s)$ for a state $s \in \mathcal{S}$ and a weight vector $w \in \mathbb{R}^d$, the value function approximant at s can be expressed as

$$V_{\phi,w}(s) = \phi(s)^\top w.$$

We write the *feature matrix* $\Phi \in \mathbb{R}^{S \times d}$ whose rows correspond to the per-state feature vectors $(\phi(s), s \in \mathcal{S})$. This leads to the more concise value function approximation

$$V_{\phi,w} = \Phi w.$$

In the classic linear function approximation literature, the feature map ϕ is held fixed, and the agent adapts only the weights w to attempt to improve its predictions. By contrast, in deep reinforcement learning, ϕ itself is parameterized by a neural network and is typically updated alongside w to improve predictions about the value function.

We measure the accuracy of the linear approximation $V_{\phi,w}$ in terms of the squared ξ -weighted l_2 norm, for $\xi \in \mathcal{P}(\mathcal{S})$,¹

$$\|V_{\phi,w} - V^\pi\|_\xi^2 = \sum_{s \in \mathcal{S}} \xi(s) (\phi(s)^\top w - V^\pi(s))^2.$$

The ξ -weighted norm describes the importance of each state.

6.2.1 Auxiliary Tasks

In deep reinforcement learning, the agent can use its representation ϕ to make additional predictions on a set of T auxiliary task functions $\{\psi_t \in \mathbb{R}^{\mathcal{S}}\}_{t \in \{1, \dots, T\}}$ where each ψ_t maps states to real values [Jaderberg et al., 2017, Bellemare et al., 2019, Dabney et al., 2021]. These predictions are used to refine the representation itself. We collect these targets into an *auxiliary task matrix* $\Psi \in \mathbb{R}^{S \times T}$ whose rows are $\psi(s) = [\psi_1(s), \dots, \psi_T(s)]$. We are interested in the case of linear task approximation

$$\hat{\Psi} = \Phi W,$$

where $W \in \mathbb{R}^{d \times T}$ is a weight matrix, and want to choose Φ and W such that $\hat{\Psi} \approx \Psi^\pi$. In this paper, we consider a variety of auxiliary tasks that ultimately

¹We assume that $\xi(s) > 0$ for all states $s \in \mathcal{S}$.

involve predicting the value functions of auxiliary *cumulants*, also referred to as general value functions [GVFs; Sutton et al., 2011]. By construction, these tasks can be decomposed into a non-zero cumulant function $g : \mathcal{S} \rightarrow \mathbb{R}^T$, mapping each state to T real values, and an expected discounted next-state term when acting according to π

$$\psi^\pi(s) = g(s) + \gamma \mathbb{E}_{S' \sim P_\pi(\cdot|s)}[\psi^\pi(S')].$$

In matrix form, this recurrence can be expressed as follows

$$\Psi^\pi = G + \gamma P_\pi \Psi^\pi = (I - \gamma P^\pi)^{-1} G,$$

where $G \in \mathbb{R}^{S \times T}$ is a *cumulant matrix* whose columns correspond to each pseudo-reward vector. An example of a family of auxiliary tasks following this structure is the successor representation [SR; Dayan, 1993]. The SR encodes a state in terms of the expected discounted time spent in other states and satisfies the following recursive form

$$\psi^\pi(s, s'') = \mathbb{I}[s = s''] + \gamma \mathbb{E}_{S' \sim P_\pi(\cdot|s)}[\psi^\pi(S', s'')],$$

for all $s'' \in \mathcal{S}$. The SR is a collection of value functions associated with the cumulant matrix $G = I$. Here we focus our analysis in its tabular form, noting that it can be extended to larger state spaces in a number of ways [Barreto et al., 2017b, Janner et al., 2020, Blier et al., 2021, Thakoor et al., 2022, Farebrother et al., 2023].

6.2.2 Monte Carlo Representations

To understand how auxiliary tasks shape representations, we start by presenting the simple case where the values of auxiliary cumulants are predicted in a supervised way. Here, the targets $\Psi^\pi = (I - \gamma P^\pi)^{-1} G$ are obtained by Monte Carlo rollouts, that is using the fixed policy to perform roll-outs and collecting the sum of rewards. The goal is to minimize the loss below

$$\mathcal{L}_{\text{aux}}^{\text{MC}}(\Phi, W) = \min_{W \in \mathbb{R}^{d \times T}} \|\Xi^{1/2}(\Phi W - \Psi^\pi)\|_F^2.$$

This method results in the network’s representation Φ , assuming a linear, fully-connected last layer, corresponding to the k principal components of the auxiliary task matrix Ψ^π if the network is other unconstrained [Bellemare et al., 2019].

Proposition 2 (Monte Carlo representations). *If $\text{rank}(\Psi^\pi) \geq d$, all representations spanning the top- d left singular vectors of Ψ^π with respect to the inner product $\langle x, y \rangle_\Xi$ are global minimizers of $\mathcal{L}_{\text{aux}}^{\text{MC}}$ and can be recovered by stochastic gradient descent.*

In large environments, it is not practical to collect full trajectories to estimate Ψ^π . Instead, practitioners learn them by bootstrapping [Sutton and Barto, 1998].

6.2.3 Temporal Difference Learning with a Deep Network

Temporal difference [TD; Sutton, 1988] is the method of choice for these auxiliary predictions. The main idea of this approach is *bootstrapping* [Sutton and Barto, 1998]. It consists in using the current estimate of the auxiliary task function to generate some targets replacing their true value Ψ^π in order to learn a new approximant of the auxiliary task function. In this paper, we consider one-step temporal difference learning where we replace the targets by a one-step prediction from the currently approximated auxiliary task function. In deep reinforcement learning, both the representation ϕ and the weights W are learnt simultaneously by minimizing the following loss function

$$\mathcal{L}_{\text{aux}}^{\text{TD}}(\phi, W) = \mathbb{E}_{\substack{s \sim \xi \\ s' \sim P_\pi(\cdot|s)}} \left[\phi(s)W - \text{SG}(g(s) + \gamma\phi(s')W) \right]^2$$

where SG denotes a *stop gradient* and means that ϕ and W are treated as a constant when taking the gradient from automatic differentiation tools [Bradbury et al., 2018, Abadi et al., 2016, Paszke et al., 2019]. Written in matrix form, we have

$$\mathcal{L}_{\text{aux}}^{\text{TD}}(\Phi, W) = \|(\Xi)^{\frac{1}{2}}(\Phi W - \text{SG}(G + \gamma P^\pi \Phi W))\|_F^2$$

Here, $\Xi \in \mathbb{R}^{S \times S}$ is a diagonal matrix with elements $\{\xi(s) : s \in \mathcal{S}\}$ on the diagonal. For clarity of exposition, we express this loss with universal value functions but the analysis can be extended to state-action values at the cost of additional complexity. The idea is to reduce the mean squared error between the approximant $\hat{\psi}$ and the target values by stochastic gradient descent (SGD). Taking the gradient of \mathcal{L} with

respect to Φ and W , we obtain the *semi-gradient* update rule

$$\begin{aligned}\Phi &\leftarrow \Phi - \alpha \Xi((I - \gamma P^\pi)\Phi W - G) W^\top \\ W &\leftarrow W - \alpha \Phi^\top \Xi((I - \gamma P^\pi)\Phi W - G)\end{aligned}\tag{6.1}$$

for a step size α . Because the values of the targets change over time, the loss \mathcal{L} does not have a proper gradient field [Dann et al., 2014] except in some particular cases [Barnard, 1993, Ollivier, 2018] and hence classic analysis of stochastic gradient descent [Bottou et al., 2018] does not apply.

6.3 Bootstrapped Representations

We now study the d -dimensional features that arise when performing value estimation of a fixed set of cumulants and how the choice of a learning method such as TD learning affects the learnt representations. Our first result characterizes representations that bootstrap themselves. We assume that the features Φ are updated in a tabular manner under the dynamics in Equation (6.1). To simplify the presentation, we now make the following invertibility assumption.

Assumption 2. *We assume that $\Phi^\top \Xi(I - \gamma P^\pi)\Phi$ is invertible for any full rank representation $\Phi \in \mathbb{R}^{S \times d}$.*

This standard assumption is for instance verified when ξ is the stationary distribution over states under π of an aperiodic, irreducible Markov chain [see e.g. Sutton et al., 2016].

An interesting characterization of the dynamical system in Equation (6.1) is its set of critical points. For a given Φ , we write

$$W_{\Phi, G}^{\text{TD}} \in \{W \in \mathbb{R}^{d \times T} \mid \nabla_W \mathcal{L}_{\text{aux}}^{\text{TD}}(\Phi, W) = 0\}.$$

Using classic linear algebra, we find that the weights W_{TD} obtained at convergence correspond to the LSTD solution [Bradtke and Barto, 1996, Boyan, 2002, Zhang et al., 2021]

$$W_{\Phi, G}^{\text{TD}} = \left(\Phi^\top \Xi(I - \gamma P^\pi)\Phi\right)^{-1} \Phi^\top \Xi G.$$

A key notion for our analysis is the concept of *invariant subspace* of a linear mapping.

Definition 13 (Gohberg et al., 2006). *A representation $\Phi \in \mathbb{R}^{S \times d}$ spans a real invariant subspace of a linear mapping $M : \mathcal{S} \rightarrow \mathbb{R}^{|\mathcal{S}|}$ if the column span of Φ is preserved by M , that is in matrix form*

$$\text{span}(M\Phi) \subseteq \text{span}(\Phi).$$

For instance, any real eigenvector of M generates one of its one-dimensional real invariant subspaces.

We are now equipped with the tools to enumerate the set of *critical representations* $\{\Phi \in \mathbb{R}^{S \times d} \mid \nabla_{\Phi} \mathcal{L}_{\text{aux}}^{\text{TD}}(\Phi, W_{\Phi}^{\text{TD}}) = 0\}$ in the lemma below.

Lemma 16 (Critical representations for TD). *All full rank representations which are critical points to $\mathcal{L}_{\text{aux}}^{\text{TD}}$ span real invariant subspaces of $(I - \gamma P^{\pi})^{-1} G G^{\top} \Xi$, that is $\text{span}((I - \gamma P^{\pi})^{-1} G G^{\top} \Xi \Phi) \subseteq \text{span}(\Phi)$.*

Proof. The proof is given in Appendix 6.C and relies on the view of LSTD as an oblique projection [Scherrer, 2010]. \square

In the particular case of an identity cumulant matrix and a uniform distribution over states, this set can be more directly expressed as the representations invariant under the transition dynamics.

Corollary 6. *If $G = I$ and $\Xi = I/|S|$, all full rank representations which are critical points to $\mathcal{L}_{\text{aux}}^{\text{TD}}$ span real invariant subspaces of the invariant subspaces of P^{π} .*

Similarly to how the top principal components of a matrix explain most of its variability [Hotelling, 1933], these critical representations are not equally informative of the dynamics of the environment.

This motivates the need to understand the behavior of the updates from Equation (6.1). To ease the analysis, we assume that the weights W have converged

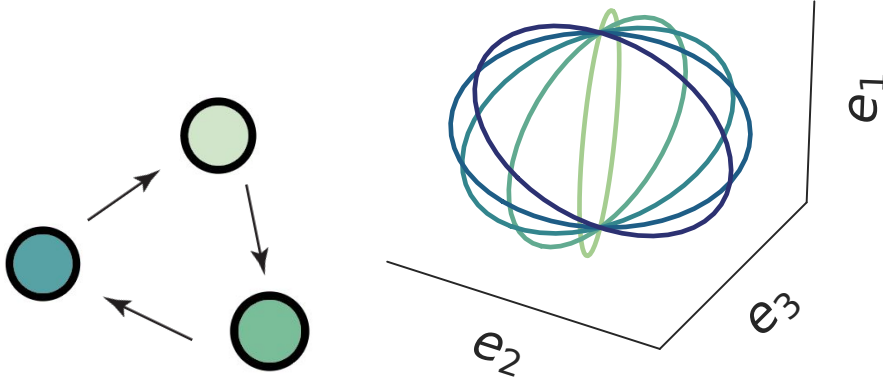


Figure 6.2: A simple 3-state MDP (left). Five subspaces, each represented by a circle, spanned by Φ during the last training steps of gradient descent on $\mathcal{L}_{\text{aux}}^{\text{TD}}$ for $d = 2$ (right).

perfectly to $W_{\Phi, G}^{\text{TD}}$ at each time step [Le Lan et al., 2023a] and consider the following continuous-time dynamics.

$$\frac{d}{dt}\Phi = -\nabla_{\Phi}\mathcal{L}(\Phi, W_{\Phi, G}^{\text{TD}}) = -F(\Phi), \quad (6.2)$$

where:

$$F(\Phi) := 2\Xi \left((I - \gamma P^{\pi})\Phi W_{\Phi, G}^{\text{TD}} - G \right) (W_{\Phi, G}^{\text{TD}})^{\top}.$$

Our key result is that the stable critical points of this ordinary differential equation correspond to the *real* top- d invariant subspace of P^{π} , when this exists.

Theorem 9 (TD representations). *Assume $G = I$, P^{π} symmetric and a uniform distribution ξ over states. Let $\lambda_1, \dots, \lambda_{|\mathcal{S}|}$ be the (possibly complex) eigenvalues of P^{π} , ordered by decreasing real part $\text{Re}(\lambda_i) \geq \text{Re}(\lambda_{i+1})$, $i \in \{1, \dots, |\mathcal{S}|\}$. If Φ is initialized to be orthogonal, under the dynamics in Equation (6.2), all real invariant subspaces of dimension d are critical points, and any non top- d real invariant subspace, if it exists, is unstable.*

The result above implies that the TD algorithm converges towards a real top- d invariant subspace or diverges with probability 1. While real diagonalisable transition matrices always induce real invariant subspaces, complex eigenvalues do not guarantee their existence and in such a case, where there is no top- d real

invariant subspace, the representation learning algorithm *does not converge*. As an illustration, consider the three-state MDP depicted in Figure 6.2, left, whose transition matrix is complex diagonalisable and given by

$$P^\pi = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Its eigenvalues are $\lambda_1 = 1$ associated to the real eigenvector e_1 and the complex conjugate pair $(\lambda_2, \bar{\lambda}_2) = (e^{2\pi i/3}, e^{-2\pi i/3})$, associated to the pair of real eigenvectors (e_2, e_3) . Hence, the real invariant subspaces of P^π are $\{0\}$, $\text{span}(e_1)$, $\text{span}(e_2, e_3)$, $\text{span}(e_1, e_2, e_3)$. Note that there is no 2-dimensional real invariant subspace containing the top eigenvector e_1 . Consequently, the 2-dimensional representation learnt by gradient descent on the TD learning rule with $G = I$ does not converge and rotates in the higher dimensional subspace $\text{span}(e_1, e_2, e_3)$ (see Figure 6.2, right).

To understand the importance of the stop-gradient in TD learning, it useful to study the representations arising from the minimization of the following loss function

$$\mathcal{L}_{\text{aux}}^{\text{res}}(\Phi, W) = \|\Xi^{\frac{1}{2}} (\Phi W - (G + \gamma P^\pi \Phi W))\|_F^2,$$

which corresponds to residual gradient algorithms [Baird, 1995]. While it has been remarked on before that the weights minimizing $\mathcal{L}_{\text{aux}}^{\text{res}}(\Phi, W)$ for a fixed Φ differ from $W_{\Phi, G}^{\text{TD}}$ [see Lemma 23; Lagoudakis and Parr, 2003, Scherrer, 2010], this objective function also has a different optimal representation

Proposition 3 (Residual representations). *Let $d \in \{1, \dots, S\}$ and F_d be the top d left singular vectors of G with respect to the inner product $\langle x, y \rangle_\Xi = y^\top \Xi x$, for all $x, y \in \mathbb{R}^{|S|}$. All representations spanning $(I - \gamma P^\pi)^{-1} F_d$ are global minimizers of $\mathcal{L}_{\text{aux}}^{\text{res}}$ and can be recovered by stochastic gradient descent.*

While TD and Monte Carlo representations are in general different, in the particular case of symmetric transition matrices and orthogonal cumulant matrices, they are the same.

MAIN ALGORITHM	l_1 -BALL OPTIMAL REPRESENTATION	REPRESENTATION	LEARNT LOSS	REPRESENTATION
BATCH MC	SVD $\left((I - \gamma P^\pi)^{-1}\right)$		MC	SVD $\left((I - \gamma P^\pi)^{-1} G\right)$
RESIDUAL	SVD $\left((I - \gamma P^\pi)^{-1}\right) \Sigma_d$		RESIDUAL	$(I - \gamma P^\pi)^{-1}$ SVD (G)
TD	Φ_{TD}^*		TD	INV $\left((I - \gamma P^\pi)^{-1} G G^\top \xi\right)$

Table 6.1: Different types of representation loss and their induced representations. The supervised targets $\Psi \in \mathbb{R}^{S \times T}$ are $(I - \gamma P^\pi)^{-1} G$. SVD(M) denotes the top- d left singular vectors of M, INV(M) the top- d invariant subspace of M and $\Sigma_d \in \mathbb{R}^{d \times d}$ the diagonal matrix with the top- d singular values of $(I - \gamma P^\pi)^{-1}$ on its diagonal.

Corollary 7 (Symmetric transition matrices). *If a cumulant matrix $G \in \mathbb{R}^{S \times T}$ (with $T \geq S$) has unit-norm, orthogonal columns (e.g. $G = I$), the representations learnt from the supervised objective $\mathcal{L}_{\text{aux}}^{\text{MC}}$ and the TD update rule $\mathcal{L}_{\text{aux}}^{\text{TD}}$ are the same for symmetric transition matrices P^π under a uniform state distribution ξ .*

This is because eigenvectors and singular vectors are identical in that setting and the eigenvalues of the successor representation are all positive.

6.4 Representations for Policy Evaluation

With the results from the previous section, the question that naturally arises is which approach results in better representations. To provide an answer, we consider a two-stage procedure. First, we learn a representation Φ by predicting the values of T auxiliary cumulants simultaneously, using one of the learning rules described in Section 6.3. Then, we retain this representation and perform policy evaluation. If the value function is estimated on-policy, it converges towards the LSTD solution [Tsitsiklis and Van Roy, 1996]

$$\hat{V}^{\text{TD}} = \Phi w_\Phi^{\text{TD}}$$

where $w_\Phi^{\text{TD}} = \left(\Phi^\top \Xi (I - \gamma P^\pi) \Phi\right)^{-1} \Phi^\top \Xi r_\pi$. We are interested in whether this value function results in low approximation error on average over random reward functions r_π , that is we want the following error to be small

$$\mathbb{E}_{r_\pi} [\|\Phi w_\Phi^{\text{TD}} - V^\pi\|_\xi^2] \quad (6.3)$$

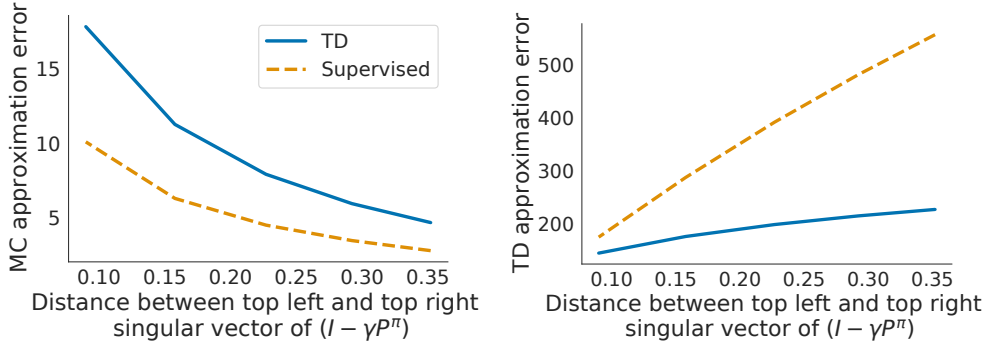


Figure 6.3: MC (left) and TD (right) approximation errors as a function of the misalignment of the top left and right singular vector of the SR induced by greedifying the policy. Trained with $\mathcal{L}_{\text{aux}}^{\text{MC}}, \mathcal{L}_{\text{aux}}^{\text{TD}}, G = I, d = 1$ on a 4-state room.

where the expectation is over the reward functions r_π sampled uniformly over the l_1 ball $\{r_\pi \in \mathbb{R}^S \mid \|r_\pi\|_1 \leq 1\}$. This set models an unknown reward function.

We say that a representation Φ_{TD}^* is l_1 -ball optimal for TD learning when it minimizes the error in Equation (6.3). Here Φ_{TD}^* depends on the transition dynamics of the environment but not on the reward function.

Lemma 17. *A representation Φ_{TD}^* is l_1 -ball optimal for TD learning iff it is a solution of the following optimization problem.*

$$\Phi_{\text{TD}}^* \in \arg \min_{\Phi} \left\| \Xi^{1/2} (\Phi W_{\Phi, I}^{\text{TD}} - (I - \gamma P^\pi)^{-1}) \right\|_F^2.$$

When P^π is symmetric and $\Xi = I/|\mathcal{S}|$, the minimum is achieved by both the top- d left singular vectors and top- d invariant subspace of the SR. However, as the misalignment between the top- d left and top- d right singular vectors of $(I - \gamma P^\pi)$ increases, the top- d invariant subspace results in lower error compared to the top- d singular vectors (see Figure 6.3); note that here, none of them achieves Φ_{TD}^* and hence $G = I$ is not l_1 -ball optimal for TD learning.

As a comparison, we study which representations are l_1 -ball optimal for linear batch Monte Carlo policy evaluation. In that setting, we are given a dataset consisting of states and their associated value, which can be estimated by the realisation of the random return [Bellemare et al., 2017, Sutton and Barto, 2018],

and the weights are learnt by least square regression. As above, we want the features minimizing

$$\mathbb{E}_{r_\pi}[\|\Phi w_\Phi^{\text{MC}} - V^\pi\|_\xi^2] \quad (6.4)$$

where $\hat{V}^{\text{MC}} = \Phi w_\Phi^{\text{MC}}$ is the value function learnt at convergence and $w_\Phi^{\text{MC}} = (\Phi^\top \Xi \Phi)^{-1} \Phi^\top \Xi V^\pi$.

Lemma 18. *A representation Φ_{MC}^* is l_1 -ball optimal for batch Monte Carlo policy evaluation if its column space spans the top- d left singular vectors (with respect to the inner product $\langle x, y \rangle_\Xi$) of $(I - \gamma P^\pi)^{-1}$.*

Unlike TD, Φ_{MC}^* is achieved by training $\mathcal{L}_{\text{aux}}^{\text{MC}}$ with $G = I$.

We summarize in Table 6.1 our representation learning results mentioned throughout Section 6.3 and Section 6.4. For completeness, we also include l_1 -ball optimal representations for residual algorithms. Proofs can be found in Appendix 6.D.

6.4.1 TD and Monte Carlo Need Different Cumulants

Having characterized which features common auxiliary tasks capture and what representations are desirable to support training the main value function, we now show that MC policy evaluation and TD learning need different cumulants. In large environments, we are interested in cumulant matrices encoding a small number of tasks $T \ll S$.

Lemma 19. *Denote B_T the top- T right singular vectors of the SR and $\mathcal{O}(T, S)$ the set of orthogonal matrices in $\mathbb{R}^{T \times S}$. Training auxiliary tasks in a MC way with any G from the set $\{G \in \mathbb{R}^{S \times T} \mid \exists M \in \mathcal{O}(T, S), G = B_T M\}$ results in an l_1 -ball optimal representation for batch Monte Carlo.*

We showed in Section 6.3 that training auxiliary tasks by TD does not always converge when the transition matrix has complex eigenvalues. Maybe surprisingly, we find that this is not problematic when learning the main value function by TD. Indeed, the rotation of its own weights balances the rotation of the underlying representation.

Lemma 20. *Let $\{\Phi_\omega\}$ be the set of rotating representations from Figure 6.2 learnt by TD learning with $G = I$ and $d = 2$. All these representations are equally good for learning the main value function by TD learning, that is $\forall \omega \in [0, 1]$,*

$$\mathbb{E}_{\|r\|_2^2 < 1} \left\| \Phi_\omega w_{\Phi_\omega}^{\text{TD}} - V^\pi \right\|_F^2$$

is constant and independent of ω .

Although $G = I$ does not always lead to Φ_{TD}^* when training $\mathcal{L}_{\text{aux}}^{\text{TD}}$, by analogy with the MC setting, we assume that $G = I$ leads to overall desirable representations. Assuming $\Xi = I/|\mathcal{S}|$, this means we would like the subspace spanned by top- d invariant subspaces of $(I - \gamma P^\pi)^{-1}$ to be the same as the subspace spanned by the top d invariant subspaces of $(I - \gamma P^\pi)^{-1} G G^\top$.

Lemma 21. *The set of cumulant matrices $G \in \mathbb{R}^{S \times T}$ that preserve the top- T invariant subspaces of the successor representation by TD learning are the top- T orthogonal invariant subspaces of $(I - \gamma P^\pi)^{-1}$, that is satisfying $G^\top G = I$ by orthogonality and $(I - \gamma P^\pi)^{-1} G \subseteq G$ by the invariance property.*

Unlike the MC case, a desirable cumulant matrix should encode the exact same information as the representation being learnt and the choice of a parametrization here matters.

6.4.2 A Deeper Analysis of Random Cumulants

We now study random cumulants which have mainly been used in the literature [Dabney et al., 2021, Lyle et al., 2021, Farebrother et al., 2023] as a heuristic to learn representations. We aim to explain their recent achievements as a pre-training technique [Farebrother et al., 2023] and their effectiveness in sparse reward environments [Lyle et al., 2021].

Proposition 4 (MC Error bound). *Let $G \in \mathbb{R}^{S \times T}$ be a sample from a standard gaussian distribution and assume $d \leq T$. Let F_d be the top- d left singular vectors of the successor representation $(I - \gamma P^\pi)^{-1}$ and \hat{F}_d be the top left singular vectors*

of $(I - \gamma P^\pi)^{-1}G$. Denote $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_S$ the singular values of the SR and $\text{dist}(F_d, \hat{F}_d)$ the $\sin \theta$ distance between the subspaces spanned by F_d and \hat{F}_d . We have

$$\mathbb{E}[\text{dist}(F_d, \hat{F}_d)] \leq \sqrt{\frac{d}{T-d-1} \frac{\sigma_{d+1}}{\sigma_d}} + \frac{e\sqrt{T}}{T-d} \left(\sum_{j=d+1}^n \frac{\sigma_j^2}{\sigma_d^2} \right)^{\frac{1}{2}}$$

Proof. A proof can be found in Appendix 6.E and follows arguments from random matrix theory. \square

This bound fundamentally depends on the ratio of the singular values σ_{d+1}/σ_d of the successor representation. As the oversampling parameter $(T-d)$ grows, the right hand side tends towards 0. In particular, for the right hand side to be less than ϵ , we need the oversampling parameter to satisfy $(T-d) \geq 1/\epsilon^2$. We investigate to which extent this result holds empirically for the TD objective in Subsection 6.4.1.

6.5 Empirical Analysis

In this section, we illustrate empirically the correctness of our theoretical characterizations from Section 6.3 and compare the goodness of different cumulants on the Four Rooms [Sutton and Barto, 2018] and Mountain Car [Moore, 1990] domains. Let $P_\Phi = \Phi(\Phi^\top \Phi)^\dagger \Phi^\top$. Here, any distance between two subspaces Φ and Φ^* is measured using the normalized subspace distance,² [Tang, 2019] defined by

$$\text{dist}(\Phi, \Phi^*) = 1 - \frac{1}{d} \cdot \text{Tr}(P_{\Phi^*} P_\Phi) \in [0, 1].$$

6.5.1 Synthetic Matrices

To begin, we train the TD, supervised and residual update rules from Section 6.3 up to convergence knowing the exact transition matrices P^π . In Figure 6.4 left and middle, we randomly sample 30 real diagonalisable matrices $P^\pi \in \mathbb{R}^{50 \times 50}$ to prevent any convergence issue from the TD update rule. In Figure 6.4 right, we generate symmetric transition matrices $P^\pi \in \mathbb{R}^{50 \times 50}$. To illustrate the theory, we run gradient descent on each learning rule by expressing the weights implicitly as a function of

²It is equivalent to the $\sin \theta$ distance up to some constant

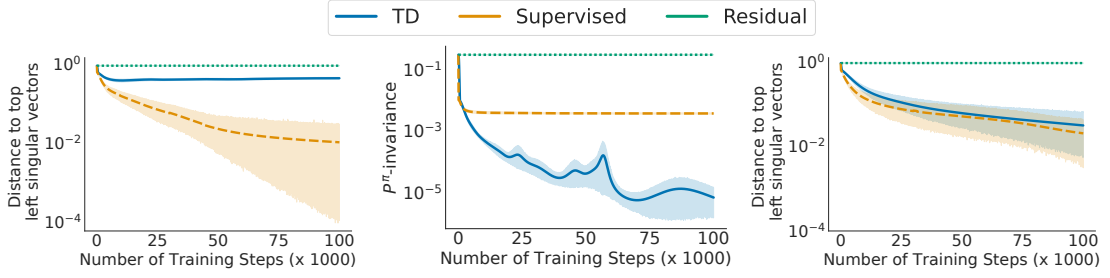


Figure 6.4: Subspace distance between Φ and the top- d left singular vectors of the SR on the left (resp. and a top- d P^π -invariant subspace in the middle) over the course of training $\mathcal{L}_{\text{aux}}^{\text{TD}}$, $\mathcal{L}_{\text{aux}}^{\text{MC}}$ and $\mathcal{L}_{\text{aux}}^{\text{res}}$ for 10^5 steps, averaged over 30 seeds ($d = 3$). MDPs with real diagonalisable (left, middle) and symmetric (right) transition matrices are randomly generated. Shaded areas represent 95% confidence intervals.

the features (see Equation (6.2) for TD for instance). Figure 6.4, left, middle show that these auxiliary task algorithms learn different representations and successfully recover our theoretical characterizations (Proposition 2, Theorem 9) from Table 6.1, right. Figure 6.4 right illustrates that the supervised and TD rules converge to the same representation for symmetric P^π , as predicted by our theory (Corollary 7).

6.5.2 Effectiveness of Random Cumulants

Following our theoretical analysis from Subsection 6.5.2, our aim is to illustrate the goodness of random cumulants at recovering the left singular vectors of the successor representation on the four room domain [Sutton et al., 1999] and to investigate to which extent an analogous result holds empirically for the TD rule. We investigate the importance of three properties of a distribution: isotropy, norm and orthogonality of the columns. We consider random cumulants from different distributions: a standard Gaussian $\mathbb{N}(0, I)$, a Gaussian distribution which columns are normalized to be unit-norm, the $O(N)$ Haar distribution and random indicators functions. Figure 6.5, left shows that the the indicator distribution which is not isotropic performs worse overall for the supervised objective and when the number of tasks is large enough, orthogonality between the columns of the cumulant matrix leads to better accuracy. In comparison, Figure 6.5, right studies the goodness of random cumulants at recovering the top- d invariant subspaces of the SR and depicts a different picture. Here, the Gaussian distribution achieves the highest error

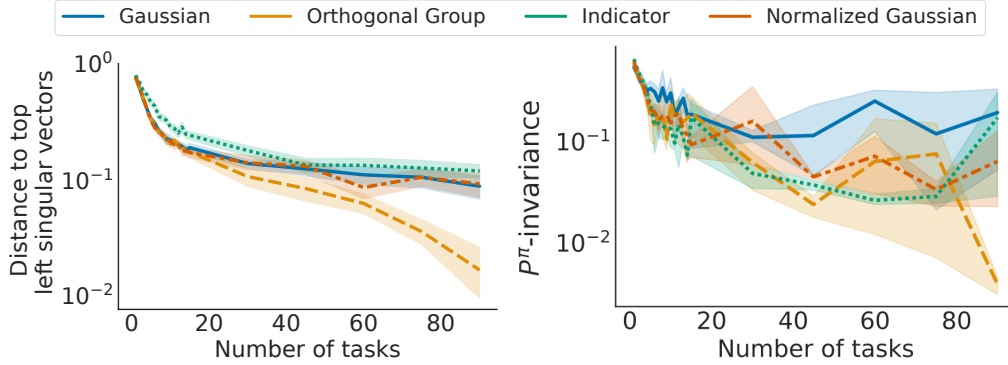


Figure 6.5: Subspace distance after 5×10^5 training steps and averaged over 30 seeds ($d = 5$) between Φ learnt with $\mathcal{L}_{\text{aux}}^{\text{MC}}$ and the top left singular vectors of the SR (left) and between Φ learnt with $\mathcal{L}_{\text{aux}}^{\text{TD}}$ and the top invariant subspaces of the SR (right) for different random cumulants, on the Four Rooms domain. Shaded areas represent estimates of 95% confidence intervals

irrespective of the number of tasks sampled while the normalized Gaussian achieves lower error suggesting the norm of the columns matter for TD training. The indicator distribution performs well for many number of sampled tasks indicating that the isotropy of the distribution is not as important for TD as it is for supervised training. Finally, the orthogonal cumulants achieve the lowest error when the number of tasks is large enough, showing this is an important property for both kinds of training.

6.5.3 Offline Pre-training

In this section we follow a similar evaluation protocol as that of Farebrother et al. [2023], but applied to the four room and Mountain car domains to allow a clear investigation of the various cumulant generation methods and the effects of their corresponding GVFs as a representation pre-training method for reinforcement learning. Details can be found in Appendix 6.A.

We consider four cumulant functions. The first two are stationary and are generated before offline pre-training begins. For ExactSVD, we compute the top- k right singular vectors of the successor representation matrix of the uniform random policy. For Normal, we generate cumulant functions sampled from a standard Normal distribution.

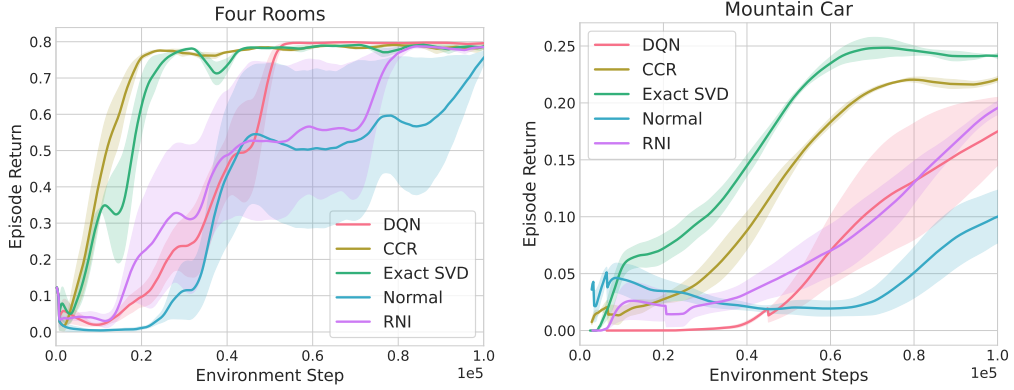


Figure 6.6: Comparing effects of offline pre-training on the Four Rooms (left) and sparse Mountain Car (right) domains for different cumulant generation methods. Results are averages over three seeds.

The second two cumulant functions are learned during offline pre-training using a separate neural network. RNI [Farebrother et al., 2023], learns a set of indicator functions which are trained to be active in a particular percentage of the states (15% in this experiment). Clustering Contrastive Representations (CCR) learns cumulants by learning a representation of the state using CPC [Oord et al., 2018], and then performs online clustering of the learned representations with k clusters. The online clustering method we use differs slightly from standard approaches in that we maintain an estimate of the frequency that each cluster center is assigned to a state, p_i , and the assigned cluster is identified with $\arg \min_i p_i \|\phi(x) - b_i\|$, where $\phi(x)$ is the learned CPC representation and b_i is cumulant i 's centroid. Examples of the cumulants produced by these four methods, and their corresponding value functions, are given in Appendix 6.A.

Figure 6.6 compares the online performance after pre-training, for various cumulant functions, with the online performance of DQN without pre-training. Two take-aways are readily apparent. First, that offline pre-training, speeds up online learning, as expected. Second, that the two best performing methods are both sensitive to the structure of the environment dynamics, directly in the case of ExactSVD and indirectly through the CPC representation for CCR.

6.6 Related Work

Optimal representations. Bellemare et al. [2019] define a notion of optimal representations for batch Monte Carlo optimization based on the worst approximation error of the value function across the set of all possible policies, later relaxed by Dabney et al. [2021]. Instead, we do not consider the control setting but focus on policy evaluation. Ghosh and Bellemare [2020] and Le Lan et al. [2022] characterize the stability, approximation and generalization errors of the SR [Dayan, 1993] and Schur representations which are P^π -invariant, a key property to ensure stability. In contrast, we formalize that predicting values functions by TD learning from $G = I$ leads to P^π -invariant subspaces.

Auxiliary tasks. Lyle et al. [2021] analyse the representations learnt by several auxiliary tasks such as random cumulants [Osband et al., 2018, Dabney et al., 2021] assuming real diagonalizability of the transition matrix P^π and constant weights W . They found that in the limit of an infinity of gaussian random cumulants, the subspace spanned by TD representations converges in distribution towards the left singular vectors of the successor representation. Instead, our theoretical analysis holds for any transition matrix and both the weights W and the features Φ are updated at each time step. Recently, Farebrother et al. [2023] rely on a random binary cumulant matrix which sparsity is controlled by means of a quantile regression loss. Finally, other auxiliary tasks regroup self-supervised learning methods [Schwarzer et al., 2021, Guo et al., 2020]. Tang et al. [2023] demonstrate that these algorithms perform an eigendecomposition of real diagonalisable transition matrix P^π , under some assumptions, suggesting a close connection to TD auxiliary tasks. Touati and Ollivier [2021], Blier et al. [2021] propose an unsupervised pretraining algorithm to learn representations based on an eigendecomposition of transition matrix P^π . They demonstrate the usefulness of their approach on discrete and continuous mazes, pixel-based MsPacman and the FetchReach virtual robot arm.

6.7 Conclusion

In this paper, we have studied representations learnt by bootstrapping methods and proved their benefit for value-based deep RL agents. Based on an analysis of the TD continuous-time dynamical system, we generalized existing work [Lyle et al., 2021] and provided evidence that TD representations are actually different from Monte Carlo representations.

Our investigation demonstrated that an identity cumulant matrix provides as much information as the TD and supervised auxiliary algorithms can carry; this work also shows that it is possible to design more compact pseudo-reward functions, though this requires prior knowledge about the transition dynamics. This led us to propose new families of cumulants which also proved useful empirically.

We assumed in this paper that the TD updates are carried out in tabular way, that is that there is not generalization between states when we update the features. An exciting opportunity for future work is to extend the theoretical results to the case where the representation is parametrized by a neural network. Other avenues for future work include scaling up the representation learning methods here introduced.

Acknowledgements

The authors would like to thank Yunhao Tang, Doina Precup and the anonymous reviewers for detailed feedback on this manuscript. We also thank Jesse Farebrother and Joshua Greaves for help with the Proto-Value Networks codebase [Farebrother et al., 2023]. Thank you to Matthieu Geist, Bruno Scherrer, Chris Dann, Diana Borsa, Remi Munos, David Abel, Daniel Guo and Bernardo Avila Pires for useful discussions too.

We also thank the Python community [Van Rossum and Drake Jr, 1995, Oliphant, 2007] for developing tools that enabled this work, including NumPy [Oliphant, 2006, Walt et al., 2011, Harris et al., 2020], SciPy [Jones et al., 2001], Matplotlib [Hunter, 2007] and JAX [Bradbury et al., 2018].

6.A Additional Empirical Results

6.A.1 Additional Details for Subsection 6.5.1

In this experiment, we selected a step size $\alpha = 0.08$ for all the algorithms. We also choose a uniform data distribution $\Xi = I/|\mathcal{S}|$ and a cumulant matrix $G = I$ for simplicity.

6.A.2 Additional Details for Subsection 6.5.2

In this experiment, we use a step size $\alpha = 5e^{-3}$ and train the different learning rules for 500k steps with 3 seeds. We consider the transition matrices induced by an epsilon greedy policy on the Four Rooms domain [Sutton et al., 1999] with $\epsilon = 0.8$ and train the supervised and TD update rules as described in Subsection 6.5.1. We provide additional empirical results in Figure 6.7.

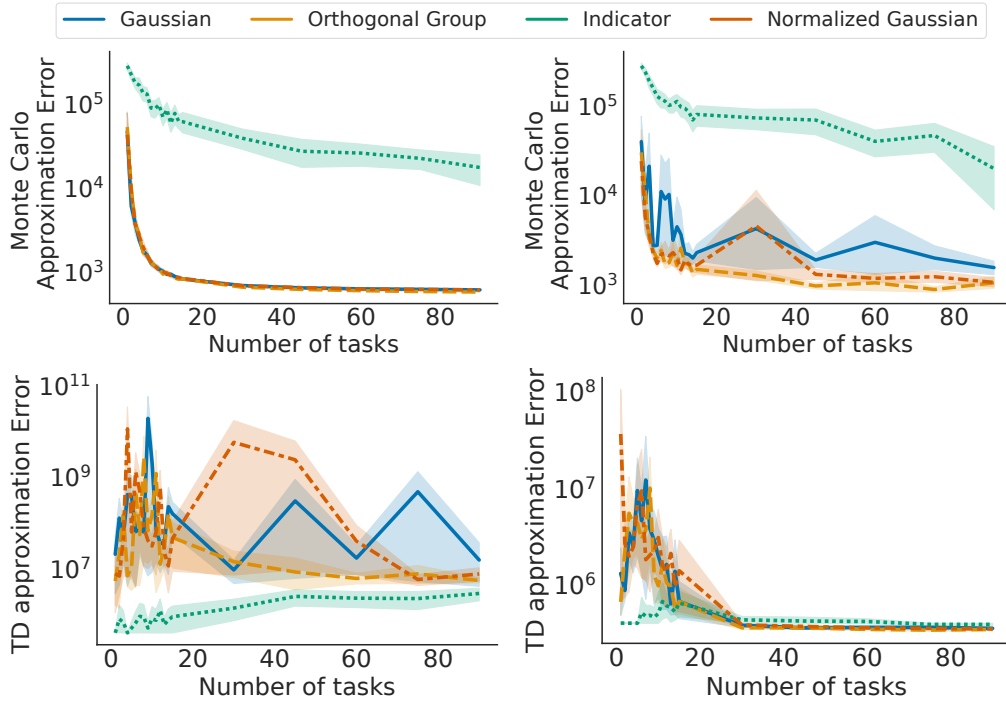


Figure 6.7: Monte Carlo and TD approximation errors after $5 \cdot 10^5$ training steps on the learning rules $\mathcal{L}_{\text{aux}}^{\text{MC}}$ (on the left column) and $\mathcal{L}_{\text{aux}}^{\text{TD}}$ (on the right column) in the Four Rooms domain for different distributions of cumulant, averaged over 30 seeds, for $d = 5$. Shaded areas represent estimates of 95% confidence intervals.

6.A.3 Additional Details for Subsection 6.5.3

Four Rooms is a tabular gridworld environment where the agent begins in a room in the top left corner and must navigate to the goal state in the lower right corner. The actions are up, down, left and right and have deterministic effects. The reward function is one upon transitioning into the goal state and zero otherwise.

Mountain Car is a two-dimensional continuous state environment where the agent must move an under-powered car from the bottom of a valley to a goal state at the top of the nearby hill. The agent observes the continuous-valued position and velocity of the car, and controls it with three discrete actions which apply positive, negative, and zero thrust to the car. In this sparse reward version of the domain the reward is one for reaching the goal and zero otherwise. In this domain, we compute the ExactSVD by first discretizing the state space into approximately 2000 states, and compute an approximate P^π by simulating transitions from uniformly random continuous states belonging to each discretized state.

In this evaluation, we first pre-train a network representation offline with a large fixed dataset produced from following the uniform random policy. During offline pre-training the agent does not observe the reward, and instead learns action-value functions, GVFs, for each of several cumulant functions. After pre-training, the GVF head is removed and replaced with a single action-value function head. This network is then trained online with DQN on the true environmental reward. Note that we allow gradients to propagate into the network representation during online training.

In the Four Rooms domain, all methods use $k = 40$ cumulants and in Mountain Car all methods use $k = 80$ cumulants.

The inputs to the network were a one-hot encoding of the observation in the Four Rooms domain and the usual position and velocity feature vector in Mountain car. The offline pre-training dataset contains 100000 and 200000 transitions for Four Rooms and Mountain Car respectively. In both cases the dataset is generated and used to fill a (fixed) replay buffer, and then the agent is trained for 400000 updates (each update using a minibatch of 32 transitions sampled uniformly from the buffer/dataset). The learning rate for both offline and online training was the

same as the standard DQN learning rate (0.00025), and similarly for the optimizer epsilon. The network architecture is a simple fully connected MLP with ReLU activations [Nair and Hinton, 2010] and two hidden layers of size 512 (first) and 256 (second), followed by a linear layer to give action-values.

We provide visualizations of the cumulants produced by each method and their corresponding value functions in Figure 6.8, Figure 6.9, Figure 6.10, Figure 6.11, Figure 6.12, Figure 6.13, Figure 6.14 and Figure 6.15.

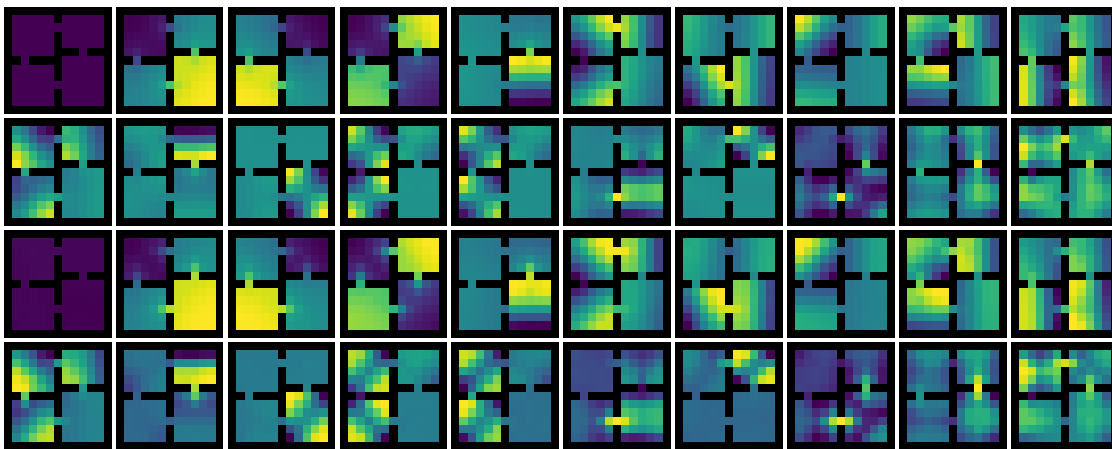


Figure 6.8: Example for ExactSVD of the learned cumulants (first two rows) and value functions (last two rows) during offline pre-training in Four Rooms under the uniform random policy.

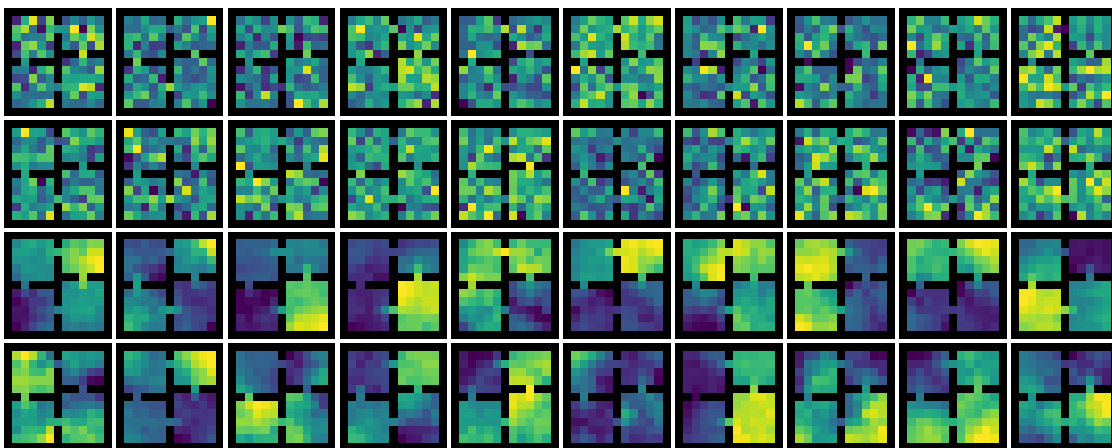


Figure 6.9: Example for Normal of the learned cumulants (first two rows) and value functions (last two rows) during offline pre-training in Four Rooms under the uniform random policy.

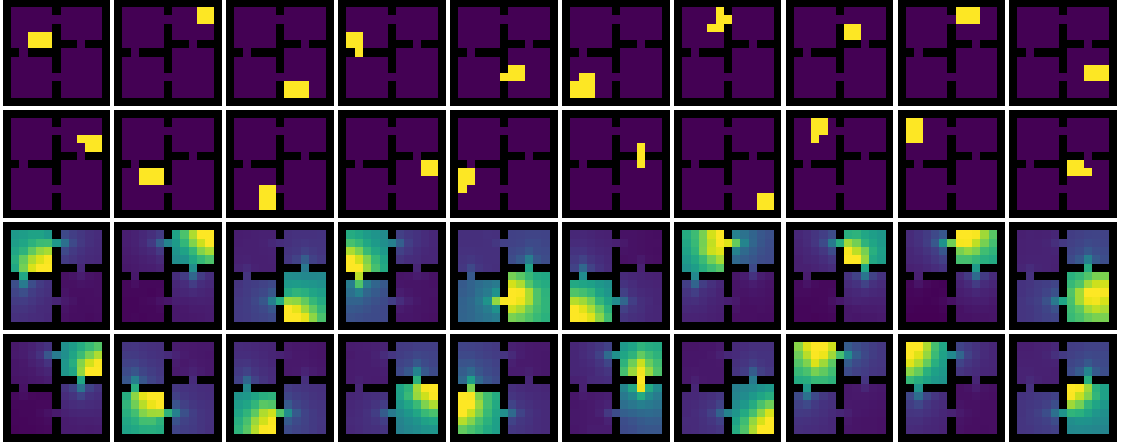


Figure 6.10: Example for CCR of the learned cumulants (first two rows) and value functions (last two rows) during offline pre-training in Four Rooms under the uniform random policy.

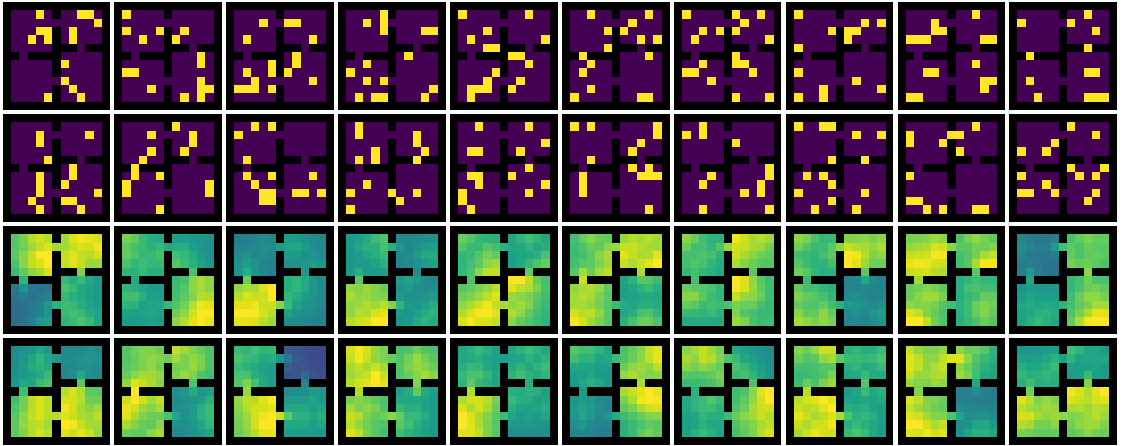


Figure 6.11: Example for RNI of the learned cumulants (first two rows) and value functions (last two rows) during offline pre-training in Four Rooms under the uniform random policy.

6.B Proofs for Section 6.2

Proposition 2 (Monte Carlo representations). *If $\text{rank}(\Psi^\pi) \geq d$, all representations spanning the top- d left singular vectors of Ψ^π with respect to the inner product $\langle x, y \rangle_\Xi$ are global minimizers of $\mathcal{L}_{\text{aux}}^{\text{MC}}$ and can be recovered by stochastic gradient descent.*

Proof. Let F_d denote the top d left singular vectors of Ψ .

$$\begin{aligned} \arg \min_{\Phi \in \mathbb{R}^{S \times d}} \min_{W \in \mathbb{R}^{d \times T}} \|\Xi^{1/2}(\Phi W - \Psi)\|_F^2 &= \arg \min_{\Phi \in \mathbb{R}^{S \times d}} \|P_{\Xi^{1/2}\Phi}^\perp \Xi^{1/2}\Psi\|_F^2 \\ &= \{\Phi \in \mathbb{R}^{S \times d} \mid \exists M \in GL_d(\mathbb{R}), \Phi = F_d M\} \end{aligned}$$

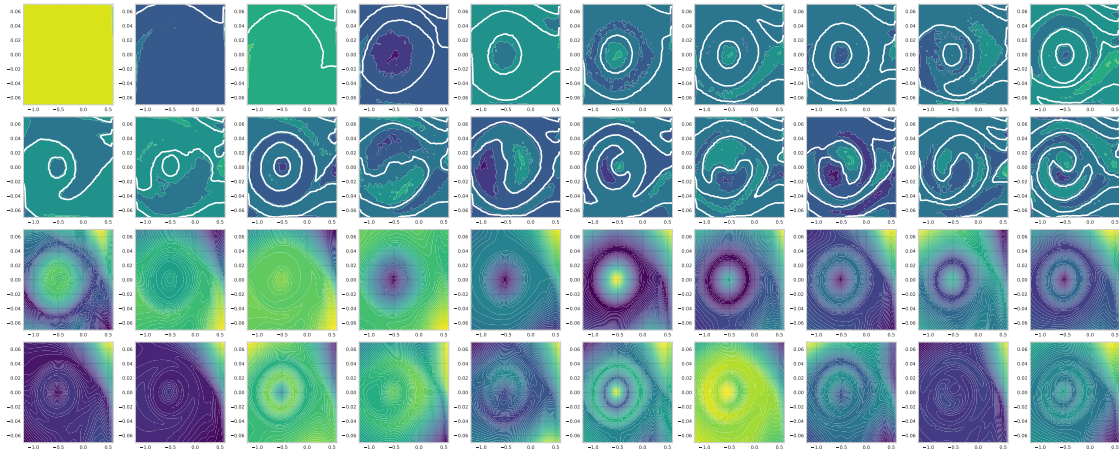


Figure 6.12: Example for ExactSVD of the learned cumulants (first two rows) and value functions (last two rows) during offline pre-training in sparse Mountain Car under the uniform random policy.

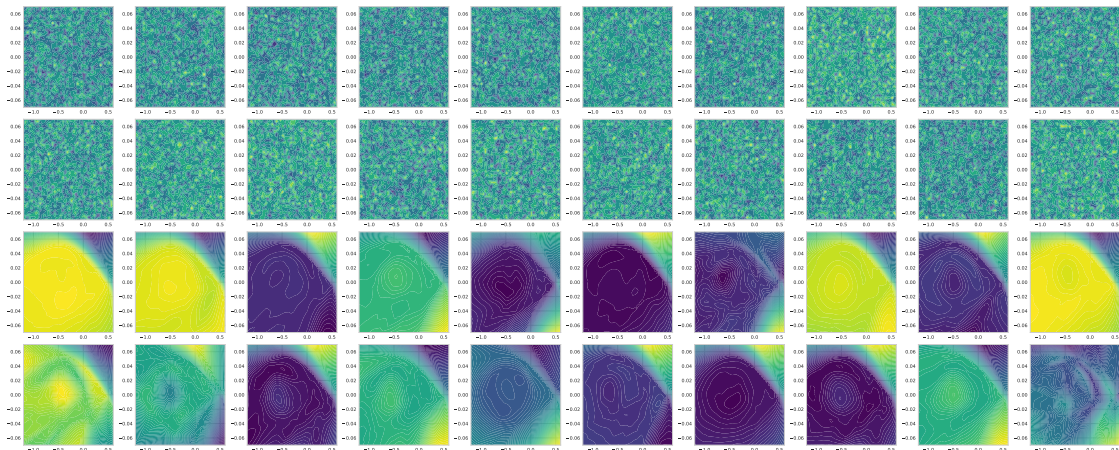


Figure 6.13: Example for Normal of the learned cumulants (first two rows) and value functions (last two rows) during offline pre-training in sparse Mountain Car under the uniform random policy.

This set of representations can be recovered by stochastic gradient descent efficiently, i.e., with number of SGD iterations scaling at most polynomially in all problem specific parameters [Ge et al., 2017, Jin et al., 2017] in the context of SGD. \square

6.C Proofs for Section 6.3

Throughout this appendix, we will use the notation $L := I - \gamma P^\pi$.

The beginning of this section is dedicated to proving the main result of Section 6.3, Theorem 9. Before that, we introduce the following necessary lemma.

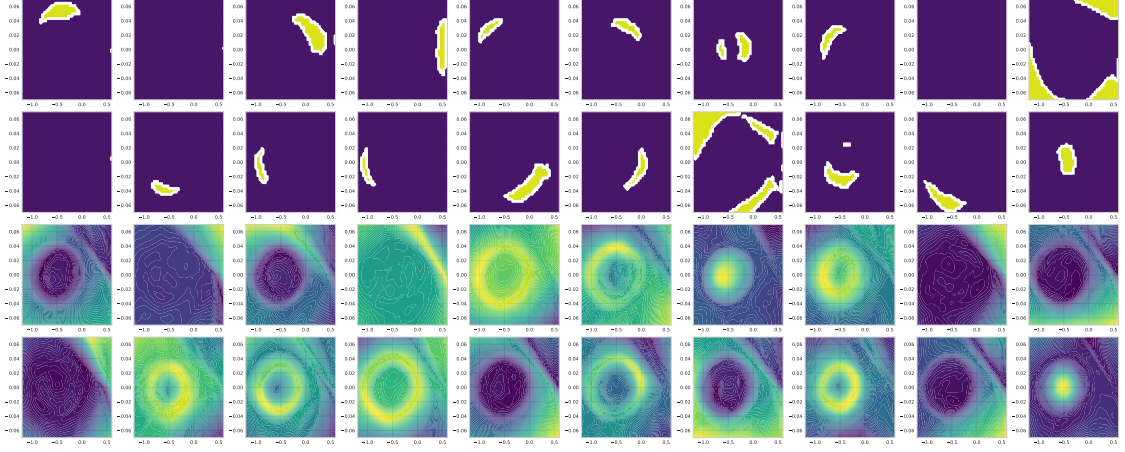


Figure 6.14: Example for CCR of the learned cumulants (first two rows) and value functions (last two rows) during offline pre-training in sparse Mountain Car under the uniform random policy.

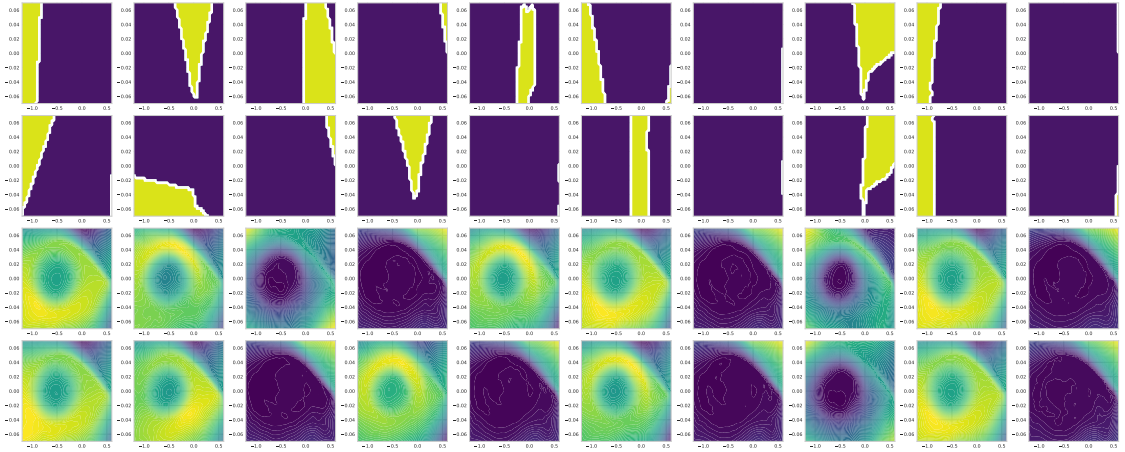


Figure 6.15: Example for RNI of the learned cumulants (first two rows) and value functions (last two rows) during offline pre-training in sparse Mountain Car under the uniform random policy.

Lemma 22. Let $\Phi \in \mathbb{R}^{S \times d}$ and $\Psi \in \mathbb{R}^{S \times T}$. Let P_Φ be a (possibly oblique) projection onto $\text{span}(\Phi)$. We have

$$P_\Phi \Psi = \Psi \iff \text{span}(\Psi) \subseteq \text{span}(\Phi)$$

Proof. P_Φ can be written as $P_\Phi = \Phi(X^\top \Phi)^{-1} X^\top$ where $\Phi, X \in \mathbb{R}^{S \times d}$ and $X^\top \Phi \in \mathbb{R}^{d \times d}$ is invertible. Write $P_\Phi = \Phi Q$ with $Q = (X^\top \Phi)^{-1} X^\top$.

(\implies) Suppose $\Psi \in \mathbb{R}^{S \times T}$ such that $P_\Phi \Psi = \Psi$. Then, $\Psi = \Phi(Q\Psi)$. Let $\omega \in \mathbb{R}^T$.

$\Psi\omega = \Phi(Q\Psi)\omega$ so $\Psi\omega \in \text{span}(\Phi)$. Hence $\text{span}(\Psi) \subseteq \text{span}(\Phi)$.

(\Leftarrow) Suppose $\text{span}(\Psi) \subseteq \text{span}(\Phi)$. Denote (e_t) the standard basis. We have $P_\Phi\Psi = (\sum_t P_\Phi(\Psi e_t)e_t^\top)$. Note that $\Psi e_t \in \text{span}(\Psi) \subseteq \text{span}(\Phi)$. Hence, there exists $y_t \in \mathbb{R}^d$ such that $\Psi e_t = \Phi y_t$. Now, $P_\Phi\Psi = (\sum_t P_\Phi(\Phi y_t)e_t^\top) = (\sum_t \Phi(X^\top\Phi)^{-1}X^\top\Phi y_t e_t^\top) = (\sum_t \Phi y_t e_t^\top) = (\sum_t \Psi e_t e_t^\top) = \Psi$. \square

Lemma 16 (Critical representations for TD). *All full rank representations which are critical points to $\mathcal{L}_{\text{aux}}^{\text{TD}}$ span real invariant subspaces of $(I - \gamma P^\pi)^{-1}GG^\top\Xi$, that is $\text{span}((I - \gamma P^\pi)^{-1}GG^\top\Xi\Phi) \subseteq \text{span}(\Phi)$.*

Proof. Start with these equations.

$$\text{For a fixed } \Phi, \nabla_W \mathcal{L}_{\text{aux}}^{\text{TD}}(\Phi, W) = 2\Phi^\top\Xi(\Phi W - G - \gamma P^\pi\Phi W)$$

$$\text{For a fixed } W, \nabla_\Phi \mathcal{L}_{\text{aux}}^{\text{TD}}(\Phi, W) = 2\Xi(\Phi W - G - \gamma P^\pi\Phi W)W^\top$$

By Assumption 2, $\Phi^\top\Xi L\Phi$ is invertible for all full rank representations Φ . Hence, for a fixed full rank Φ ,

$$\nabla_W \|(\Xi)^{\frac{1}{2}}(\Phi W - G - \gamma P^\pi \text{sg}[\Phi W])\|_F^2 = 0 \iff W_\Phi^* = (\Phi^\top\Xi L\Phi)^{-1} \Phi^\top\Xi G$$

Using the second fixed-point equation:

$$0 = (L\Phi W - G)W^\top \iff L\Phi W W^\top = G W^\top.$$

Now plugging in the expression for W_Φ^* ,

$$\begin{aligned} L\Phi (\Phi^\top\Xi L\Phi)^{-1} \Phi^\top\Xi G \left((\Phi^\top\Xi L\Phi)^{-1} \Phi^\top\Xi G \right)^\top &= G \left((\Phi^\top\Xi L\Phi)^{-1} \Phi^\top\Xi G \right)^\top \\ \iff L\Phi (\Phi^\top\Xi L\Phi)^{-1} \Phi^\top\Xi G G^\top \Xi \Phi (\Phi^\top\Xi L\Phi)^{-\top} &= G G^\top \Xi \Phi (\Phi^\top\Xi L\Phi)^{-\top} \\ \iff \Phi (\Phi^\top\Xi L\Phi)^{-1} \Phi^\top\Xi G G^\top \Xi \Phi &= L^{-1} G G^\top \Xi \Phi \\ \iff \Pi_{L^\top\Xi\Phi} L^{-1} G G^\top \Xi \Phi &= L^{-1} G G^\top \Xi \Phi \end{aligned}$$

where $\Pi_X = \Phi(X^\top\Phi)^{-1}X^\top$ is the oblique projection onto $\text{span}(\Phi)$ orthogonally to $\text{span}(X)$. This is equivalent to $\Pi_{L^\top\Xi\Phi}^\perp L^{-1}GG^\top\Xi\Phi = 0$, which is equivalent to saying that $\text{span}(\Phi)$ must be an *invariant subspace* of $L^{-1}GG^\top\Xi$ by Lemma 22.

In other words, we have shown that all non-degenerate full-rank Φ which are critical points span invariant subspaces of $L^{-1}GG^T\Xi$. We can enumerate these via the *real Jordan normal form* of $L^{-1}GG^T\Xi$. Each block of the real Jordan normal form corresponds to an invariant subspace of $L^{-1}GG^T\Xi$. Suppose that the real Jordan block sizes of $L^{-1}GG^T\Xi$ are n_1, n_2, \dots, n_b ($L^{-1}GG^T\Xi$ has b real Jordan blocks), and suppose $L^{-1}GG^T\Xi = SJS^{-1}$ is the real Jordan decomposition, with $J = \text{blkdiag}(J_{n_1}(\lambda_1), \dots, J_{n_b}(\lambda_b))$. Partition the columns of S into S_1, \dots, S_b . Then if $\Phi \in \mathbb{R}^{S \times k}$, the set of non-degenerate stationary full rank representations is:

$$\{[S_{i_1} \ \dots \ S_{i_\ell}] \mid n_{i_1} + \dots + n_{i_\ell} = k\}.$$

□

Corollary 6. *If $G = I$ and $\Xi = I/|S|$, all full rank representations which are critical points to $\mathcal{L}_{\text{aux}}^{\text{TD}}$ span real invariant subspaces of the invariant subspaces of P^π .*

Proof. Let $G = I$ and $\Xi = I/|S|$. By Lemma 16, all full rank representations which are critical points of $\mathcal{L}_{\text{aux}}^{\text{TD}}$ span real invariant subspaces of $(I - \gamma P^\pi)^{-1}$.

Let Φ be a representation spanning an invariant subspace of $(I - \gamma P^\pi)^{-1}$. By definition, $\text{span}((I - \gamma P^\pi)^{-1}\Phi) \subseteq \text{span}(\Phi)$. Because $(I - \gamma P^\pi)$ is invertible, we have $\dim((I - \gamma P^\pi)^{-1}\Phi) = \dim(\Phi)$. Hence, we actually have $\text{span}((I - \gamma P^\pi)^{-1}\Phi) = \text{span}(\Phi)$. There exists $w_1, w_2 \in \mathbb{R}^d$ such that $\Phi w_1 = (I - \gamma P^\pi)^{-1}\Phi w_2$ so $(I - \gamma P^\pi)\Phi w_1 = \Phi w_2$. It follows that $\Phi \frac{(w_1 - w_2)}{\gamma} = P^\pi \Phi w_1$. Hence, $P^\pi \Phi w_1 \in \text{span}(\Phi)$ and $\text{span}(P^\pi \Phi) \subseteq \text{span}(\Phi)$. We conclude that Φ spans an invariant subspace of P^π . □

Theorem 9 (TD representations). *Assume $G = I$, P^π symmetric and a uniform distribution ξ over states. Let $\lambda_1, \dots, \lambda_{|S|}$ be the (possibly complex) eigenvalues of P^π , ordered by decreasing real part $\text{Re}(\lambda_i) \geq \text{Re}(\lambda_{i+1})$, $i \in \{1, \dots, |S|\}$. If Φ is initialized to be orthogonal, under the dynamics in Equation (6.2), all real invariant subspaces of dimension d are critical points, and any non top- d real invariant subspace, if it exists, is unstable.*

Proof. Consider this objective:

$$\mathcal{L}(\Phi) = \frac{1}{2} \|(\Xi^{\frac{1}{2}})(\Phi W_{\Phi,G}^{\text{TD}} - G - \gamma P^\pi \text{SG}[\Phi W_{\Phi,G}^{\text{TD}}])\|_F^2,$$

and $W_{\Phi,G}^{\text{TD}} = (\Phi^\top \Xi L \Phi)^{-1} \Phi^\top \Xi G$ and define $L := I - \gamma P^\pi$. Observe that:

$$\text{For a fixed } W, \nabla_\Phi \|\Phi W - G - \gamma P^\pi \text{SG}[\Phi W]\|_F^2 = 2\Xi(L\Phi W - G)(W)^\top$$

So now we consider the continuous time dynamics:

$$\frac{d}{dt}\Phi = -\nabla_\Phi \mathcal{L}(\Phi) := -F(\Phi), \quad (6.5)$$

where:

$$F(\Phi) := \Xi(L\Phi W_{\Phi,G}^{\text{TD}} - G)(W_{\Phi,G}^{\text{TD}})^\top = \Xi L(\Pi_{L^\top \Xi \Phi} - I)L^{-1}GG^\top \Xi \Phi(\Phi^\top \Xi L \Phi)^{-\top}$$

Consider the case $G = I$ and $\Xi = I/|\mathcal{S}|$. The proof strategy consists in constructing an eigenvector $\Delta \in \mathbb{R}^{S \times d}$ of $\partial_\Phi F(\Phi)$ as a function of Φ, L, G such that $\partial_\Phi F(\Phi)[\Delta] = -\lambda \Delta$ for some $\text{Re}(\lambda) > 0$. For every non top- d invariant subspace, we prove that the Jacobian of the dynamics $-F$ has a positive real part eigenvalue.

Let Φ be a stationary point which columns are orthogonal such that $\Phi^\top \Phi = I$. Φ is an invariant subspace of P^π . Assume that Φ does not contain any of the eigenvectors corresponding to the top d eigenvalues. Define $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_d)$ its associated eigenvalues assumed distinct. We have $P\Phi = \Phi\Lambda$. Hence, $(I - \gamma P^\pi)\Phi = \Phi(I - \gamma\Lambda)$. Let λ_{\max} the largest eigenvalue of P^π not contained in Φ and let $i \in \{1, \dots, d\}$ be the largest index such that $\lambda_i < \lambda_{\max}$. Let Δ be the matrix with the eigenvector corresponding to the eigenvalue λ_{\max} in its i -th column and 0 elsewhere.

$$\begin{aligned} \partial_\Phi W_\Phi^*[\Delta] &= -(\Phi^\top L \Phi)^{-1}(\Delta^\top L \Phi + \Phi^\top L \Delta)(\Phi^\top L \Phi)^{-1} \Phi^\top G + (\Phi^\top L \Phi)^{-1} \Delta^\top G \\ &= -(\Phi^\top \Phi(I - \gamma\Lambda))^{-1}(\Delta^\top \Phi(I - \gamma\Lambda)) \\ &\quad + (1 - \gamma\lambda_{\max})\Phi^\top \Delta(\Phi^\top \Phi(I - \gamma\Lambda))^{-1} \Phi^\top G + (\Phi^\top \Phi(I - \gamma\Lambda))^{-1} \Delta^\top \\ &= (I - \gamma\Lambda)^{-1}(\Phi^\top \Phi)^{-1} \Delta^\top \text{ as } \Delta^\top \Phi = 0 \end{aligned}$$

$$\begin{aligned}
\partial_{\Phi} F(\Phi)[\Delta] &= (L\Delta W_{\Phi}^* + L\Phi(dW_{\Phi}^*)) (W_{\Phi}^*)^{\top} + (L\Phi W_{\Phi}^* - G)(dW_{\Phi}^*)^{\top} \\
&= (1 - \gamma\lambda_{\max})\Delta(I - \gamma\Lambda)^{-2}(\Phi^{\top}\Phi)^{-1} \\
&\quad + L\Phi(I - \gamma\Lambda)^{-1}(\Phi^{\top}\Phi)^{-1}\Delta^{\top}\Phi(\Phi^{\top}L\Phi)^{-\top} \\
&\quad + L\Phi(\Phi^{\top}L\Phi)^{-1}\Phi^{\top}\Delta(\Phi^{\top}\Phi)^{-1}(I - \gamma\Lambda)^{-\top} - \Delta(\Phi^{\top}\Phi)^{-1}(I - \gamma\Lambda)^{-\top} \\
&= \Delta(1 - \gamma\lambda_{\max})(I - \gamma\Lambda)^{-2}(\Phi^{\top}\Phi)^{-1} - \Delta(\Phi^{\top}\Phi)^{-1}(I - \gamma\Lambda)^{-\top} \\
&= \Delta(1 - \gamma\lambda_{\max})(I - \gamma\Lambda)^{-2} - \Delta(I - \gamma\Lambda)^{-1} \\
&= \gamma\Delta(-\lambda_{\max}I + \Lambda)(I - \gamma\Lambda)^{-2} \\
&= \gamma\Delta(-\lambda_{\max} + \lambda_i)(1 - \gamma\lambda_i)^{-2} < 0
\end{aligned}$$

Hence, any non top- d invariant subspace is unstable for gradient descent. \square

Proposition 3 (Residual representations). *Let $d \in \{1, \dots, S\}$ and F_d be the top d left singular vectors of G with respect to the inner product $\langle x, y \rangle_{\Xi} = y^{\top}\Xi x$, for all $x, y \in \mathbb{R}^{|S|}$. All representations spanning $(I - \gamma P^{\pi})^{-1}F_d$ are global minimizers of $\mathcal{L}_{\text{aux}}^{\text{res}}$ and can be recovered by stochastic gradient descent.*

Proof. We can write the loss function to be minimized as

$$\begin{aligned}
J(\Phi) &= \min_{W \in \mathbb{R}^{d \times T}} \|\Xi^{1/2}(\Phi W - (G + \gamma P^{\pi}\Phi W))\|_F^2 \\
&= \min_{W \in \mathbb{R}^{d \times T}} \|\Xi^{1/2}(\Phi W - \gamma P^{\pi}\Phi W - G)\|_F^2 \\
&= \min_{W \in \mathbb{R}^{d \times T}} \|\Xi^{1/2}((I - \gamma P^{\pi})\Phi W - G)\|_F^2
\end{aligned}$$

Now,

$$\begin{aligned}
&\arg \min_{\Phi \in \mathbb{R}^{S \times d}} \min_{W \in \mathbb{R}^{d \times T}} \|\Xi^{1/2}((I - \gamma P^{\pi})\Phi W - G)\|_F^2 \\
&= \arg \min_{\Phi \in \mathbb{R}^{S \times d}} \|P_{\Xi^{1/2}(I - \gamma P^{\pi})\Phi}^{\perp} \Xi^{1/2} G\|_F^2 \\
&= \{\Phi \in \mathbb{R}^{S \times d} \mid \Phi = (I - \gamma P^{\pi})^{-1}F_d M, M \in GL_d(\mathbb{R})\}
\end{aligned}$$

This set of representations can be recovered by stochastic gradient descent efficiently, i.e., with number of SGD iterations scaling at most polynomially in all problem specific parameters [Ge et al., 2017, Jin et al., 2017] in the context of SGD. \square

Corollary 7 (Symmetric transition matrices). *If a cumulant matrix $G \in \mathbb{R}^{S \times T}$ (with $T \geq S$) has unit-norm, orthogonal columns (e.g. $G = I$), the representations learnt from the supervised objective $\mathcal{L}_{\text{aux}}^{\text{MC}}$ and the TD update rule $\mathcal{L}_{\text{aux}}^{\text{TD}}$ are the same for symmetric transition matrices P^π under a uniform state distribution ξ .*

Proof. Assume that P^π is symmetric so that L and L^{-1} are also symmetric.

By Proposition 2, running SGD on the supervised objective $\mathcal{L}_{\text{aux}}^{\text{MC}}$ using $\Psi = L^{-1}G$ as targets results in a representation spanning the top- d left singular vectors of $L^{-1}G$ which are the same as the top- d left singular vectors of L^{-1} .

By assumption G is orthogonal, hence $GG^\top = I$. Because $L^{-1}GG^\top$ is symmetric, all its eigenvalues are real. By Theorem 9, running gradient descent on $\mathcal{L}_{\text{aux}}^{\text{TD}}$ using G as the cumulant matrix converges to the top- d eigenvectors of $L^{-1}GG^\top = L^{-1}$. Indeed, the subspaces given by the span of the right eigenvectors of L^{-1} are the only L^{-1} -invariant subspaces. These eigenvectors are also the singular vectors of L^{-1} as this matrix is symmetric.

Because P is a row stochastic matrix, we have that the spectral radius of P satisfies $\rho(P) = 1$, and therefore $\lambda(P) \subseteq [-1, 1]$. Hence:

$$\frac{1}{1 - \gamma\lambda} \in [1/(1 + \gamma), 1/(1 - \gamma)].$$

Hence, the eigenvalues of L^{-1} are positive. Because L^{-1} is symmetric, the singular values of L^{-1} are exactly its eigenvalues. Hence, the top- d eigenvectors are the top- d singular vectors and the conclusion follows. \square

6.D Proofs for Section 6.4

Lemma 17. *A representation Φ_{TD}^* is l_1 -ball optimal for TD learning iff it is a solution of the following optimization problem.*

$$\Phi_{\text{TD}}^* \in \arg \min_{\Phi} \left\| \Xi^{1/2} (\Phi W_{\Phi, I}^{\text{TD}} - (I - \gamma P^\pi)^{-1}) \right\|_F^2.$$

Proof. By definition, a representation is enough for TD learning when it is a minimizer of Equation (6.3), that is,

$$\Phi_{\text{TD}}^* \in \arg \min_{\Phi \in \mathbb{R}^{S \times d}} \mathbb{E}_{r_\pi} \|\Phi w_\Phi^{\text{TD}} - V^\pi\|_\xi^2, \quad (6.6)$$

where the expectation is over the reward functions r_π sampled uniformly over the l_1 ball $\|r_\pi\|_1^2 \leq 1$ and

$$w_\Phi^{\text{TD}} = \left(\Phi^\top \Xi (I - \gamma P^\pi) \Phi \right)^{-1} \Phi^\top \Xi r_\pi.$$

Write $P_{L^\top \Xi \Phi}^\perp = I - P_{L^\top \Xi \Phi}$ and $P_X = \Phi (X^\top \Phi)^{-1} X^\top$ the oblique projection onto $\text{span}(\Phi)$ orthogonally to $\text{span}(X)$. We have

$$\begin{aligned} \mathbb{E}_{\|r\|_1^2 \leq 1} \|\Phi w_\Phi^{\text{TD}} - V^\pi\|_\xi^2 &= \mathbb{E}_{\|r\|_1^2 \leq 1} \|\Xi^{1/2} P_{L^\top \Xi \Phi}^\perp (I - \gamma P^\pi)^{-1} r\|_2^2 \\ &= \mathbb{E}_{\|r\|_1^2 \leq 1} \|\Xi^{1/2} P_{L^\top \Xi \Phi}^\perp (I - \gamma P^\pi)^{-1} r\|_2^2 \\ &= \mathbb{E}_{\|r\|_1^2 \leq 1} \text{tr}(r^\top L^{-\top} (P_{L^\top \Xi \Phi}^\perp)^\top \Xi P_{L^\top \Xi \Phi}^\perp L^{-1} r) \\ &= \text{tr}(L^{-\top} (P_{L^\top \Xi \Phi}^\perp)^\top \Xi P_{L^\top \Xi \Phi}^\perp L^{-1} \mathbb{E}(r r^\top)) \\ &\propto \|\Xi^{1/2} P_{L^\top \Xi \Phi}^\perp L^{-1}\|_F^2 \\ &\propto \|\Xi^{1/2} (\Phi W_{\Phi, I}^{\text{TD}} - (I - \gamma P^\pi)^{-1})\|_F^2 \end{aligned}$$

The penultimate line comes from the fact that r is sampled from an isotropic distribution. \square

Lemma 18. *A representation Φ_{MC}^* is l_1 -ball optimal for batch Monte Carlo policy evaluation if its column space spans the top- d left singular vectors (with respect to the inner product $\langle x, y \rangle_\Xi$) of $(I - \gamma P^\pi)^{-1}$.*

Proof. We have

$$\begin{aligned} \mathbb{E}_{\|r\|_1^2 \leq 1} \|\hat{V}^{\text{MC}} - V^\pi\|_\xi^2 &= \mathbb{E}_{\|r\|_1^2 \leq 1} \|P_{\Xi^{1/2} \Phi}^\perp \Xi^{1/2} (I - \gamma P^\pi)^{-1} r\|_2^2 \\ &= \mathbb{E}_{\|r\|_1^2 \leq 1} \text{tr}(r^\top L^{-\top} \Xi^{1/2} P_{\Xi^{1/2} \Phi}^\perp \Xi^{1/2} L^{-1} r) \\ &= \text{tr}(L^{-\top} \Xi^{1/2} P_{\Xi^{1/2} \Phi}^\perp \Xi^{1/2} L^{-1} \mathbb{E}(r r^\top)) \\ &= \|P_{\Xi^{1/2} \Phi}^\perp \Xi^{1/2} L^{-1}\|_F^2 \end{aligned}$$

Write $(I - \gamma P^\pi)^{-1} = F\Sigma B^\top$ the weighted SVD of $(I - \gamma P^\pi)^{-1}$ where $F \in \mathbb{R}^{S \times S}$ such that $F^\top \Xi F = I$ and $B \in \mathbb{R}^{S \times S}$ such that $B^\top B = I$. Write F_d the top- d left singular vectors corresponding to the top- d singular values on the diagonal of Σ . By definition, an l_1 -ball optimal representation is solution to the following optimization problem

$$\begin{aligned} \arg \min_{\Phi \in \mathbb{R}^{S \times d}} \mathbb{E}_{\|r\|_1 \leq 1} \|\hat{V}^{\text{MC}} - V^\pi\|_\xi^2 &= \arg \min_{\Phi \in \mathbb{R}^{S \times d}} \|P_{\Xi^{1/2}\Phi}^\perp \Xi^{1/2} L^{-1}\|_F^2 \\ &= \arg \min_{\Phi \in \mathbb{R}^{S \times d}} \|P_{\Xi^{1/2}\Phi}^\perp \Xi^{1/2} F\Sigma B^\top\|_F^2 \end{aligned}$$

By the Eckart-Young theorem, $\|P_{F_d}^\perp \Xi^{1/2} F\Sigma B^\top\|_F^2 \leq \|P_\Phi^\perp \Xi^{1/2} F\Sigma B^\top\|_F^2$. Hence, the set of optimal representations is $\{F_d M, M \in GL_d(\mathbb{R})\}$. \square

Lemma 23. Write $F_d \Sigma_d B_d^\top$ the truncated weighted SVD of the successor representation $(I - \gamma P^\pi)^{-1}$. A representation is l_1 -ball optimal for residual policy evaluation if its column space spans $F_d \Sigma_d$.

Proof. Write $(I - \gamma P^\pi)^{-1} = F\Sigma B^\top$ the weighted SVD of $(I - \gamma P^\pi)^{-1}$ where $F \in \mathbb{R}^{S \times S}$ such that $F^\top \Xi F = I$ and $B \in \mathbb{R}^{S \times S}$ such that $B^\top B = I$. Write F_d the top- d left singular vectors corresponding to the top- d singular values on the diagonal of Σ . For a fixed $\Phi \in \mathbb{R}^{S \times d}$, the solution of $\min_{w \in \mathbb{R}^d} \|\Xi^{1/2}(\Phi w - (r_\pi + \gamma P^\pi \Phi w))\|_F^2$ is the Bellman residual minimizing approximation [Lagoudakis and Parr, 2003] and is given by

$$w_\Phi^{\text{res}} = \left((\Phi - \gamma P^\pi \Phi)^\top \Xi (\Phi - \gamma P^\pi \Phi) \right)^{-1} (\Phi - \gamma P^\pi \Phi)^\top \Xi r_\pi.$$

Hence, the value approximant can be expressed by means of an orthogonal projection matrix as follows

$$\Phi w_\Phi^{\text{res}} = (I - \gamma P^\pi)^{-1} \Xi^{-1/2} P_{\Xi^{1/2}(I - \gamma P^\pi)\Phi} \Xi^{1/2} r_\pi$$

where $P_X = X(X^\top X)^{-1}X^\top$ denotes an orthogonal projection. By definition, a representation l_1 -ball optimal for residual policy evaluation is solution to the

following optimization problem

$$\begin{aligned}
& \arg \min_{\Phi \in \mathbb{R}^{S \times d}} \mathbb{E}_{\|r\|_1^2 \leq 1} \|\hat{V}^{\text{res}} - V^\pi\|_\xi^2 \\
&= \arg \min_{\Phi \in \mathbb{R}^{S \times d}} \|\Xi^{1/2}(I - \gamma P^\pi)^{-1} \Xi^{-1/2} P_{\Xi^{1/2}(I - \gamma P^\pi)\Phi} \Xi^{1/2} r_\pi - \Xi^{1/2}(I - \gamma P^\pi)^{-1} r_\pi\|_F^2 \\
&= \arg \min_{\Phi \in \mathbb{R}^{S \times d}} \|\Xi^{1/2}(I - \gamma P^\pi)^{-1} \Xi^{-1/2} P_{\Xi^{1/2}(I - \gamma P^\pi)\Phi} \Xi^{1/2} - \Xi^{1/2}(I - \gamma P^\pi)^{-1}\|_F^2 \\
&= \arg \min_{\Phi \in \mathbb{R}^{S \times d}} \|\Xi^{1/2}(I - \gamma P^\pi)^{-1} P_{\Xi^{1/2}(I - \gamma P^\pi)\Phi}^\perp\|_F^2
\end{aligned}$$

Using an oblique projection,

$$\Phi w_\Phi^{\text{res}} = (I - \gamma P^\pi)^{-1} \Xi^{-1/2} P_{\Xi^{1/2}(I - \gamma P^\pi)\Phi} \Xi^{1/2} r_\pi$$

$$\begin{aligned}
& \arg \min_{\Phi \in \mathbb{R}^{S \times d}} \mathbb{E}_{\|r\|_1^2 \leq 1} \|\hat{V}^{\text{res}} - V^\pi\|_\xi^2 \\
&= \arg \min_{\Phi \in \mathbb{R}^{S \times d}} \|\Xi^{1/2}(I - \gamma P^\pi)^{-1} \Xi^{-1/2} P_{\Xi^{1/2}(I - \gamma P^\pi)\Phi} \Xi^{1/2} r_\pi - \Xi^{1/2}(I - \gamma P^\pi)^{-1} r_\pi\|_F^2 \\
&= \arg \min_{\Phi \in \mathbb{R}^{S \times d}} \|\Xi^{1/2}(I - \gamma P^\pi)^{-1} \Xi^{-1/2} P_{\Xi^{1/2}(I - \gamma P^\pi)\Phi} \Xi^{1/2} - \Xi^{1/2}(I - \gamma P^\pi)^{-1}\|_F^2 \\
&= \arg \min_{\Phi \in \mathbb{R}^{S \times d}} \|\Xi^{1/2}(I - \gamma P^\pi)^{-1} P_{\Xi^{1/2}(I - \gamma P^\pi)\Phi}^\perp\|_F^2
\end{aligned}$$

$$L^{-1} = U \Sigma V^\top$$

$L^{-1} \times$ the top d right singular vectors of $(I - \gamma P^\pi)^{-1}$ is a solution. Let U_d, Σ_d, V_d correspond to the top d svals. Lets say that U_d is $S \times d$, Σ_d is square, and V_d is also $S \times d$. What is $V^\top V_d = \begin{bmatrix} I_d \\ 0 \end{bmatrix}$.

We want $L\Phi = V_d$ so $\Phi = L^{-1}V_d = U\Sigma V^\top V_d = U_d \Sigma_d$. If $L\Phi = V_d$, then $P_{L\Phi}^\perp = P_{V_d}^\perp$, so $L^{-1}P_{V_d}^\perp = U_d^\perp \Sigma_d^\perp (V_d^\perp)^\top$, so the objective is now sum of the last $(S - d)$ singular values squared.

□

6.E Proofs for Subsection 6.4.1

Lemma 19. Denote B_T the top- T right singular vectors of the SR and $\mathcal{O}(T, S)$ the set of orthogonal matrices in $\mathbb{R}^{T \times S}$. Training auxiliary tasks in a MC way with any

G from the set $\{G \in \mathbb{R}^{S \times T} | \exists M \in \mathcal{O}(T, S), G = B_T M\}$ results in an l_1 -ball optimal representation for batch Monte Carlo.

Proof. By Lemma 18, a representation is l_1 -ball optimal for batch Monte Carlo policy evaluation if it spans the top- d left singular vectors of the successor representation.

Let $G \in \mathbb{R}^{S \times T}$ be a cumulant matrix.

$$\mathcal{L}_{\text{aux}}^{\text{SL}}(\Phi) = \min_{W \in \mathbb{R}^{d \times S}} \|(\Phi W - (I - \gamma P^\pi)^{-1} G)\|_F^2$$

By Proposition 2, we know that training on such a loss with $G = I$ results in a representation spanning the same subspace as the left singular vectors of the SR, that is $\{\Phi \in \mathbb{R}^{S \times d} | \exists M \in GL_d(\mathbb{R}), \Phi = F_d M\}$ where F_d are the left singular vectors of the SR. We note that there is not a unique matrix G resulting into a representation spanning that subspace. In particular, training with any of the matrices from the set of cumulant matrices $\mathcal{G}(G) = \{G' \in \mathbb{R}^{S \times T} | \exists M \in \mathcal{O}(T, S), G' = GM\}$ results in the same representation, where $\mathcal{O}(T, S)$ denotes the set of orthogonal matrices in $\mathbb{R}^{T \times S}$ (rows have l_2 norm 1).

We are interested in finding a cumulant matrix $G \in \mathbb{R}^{S \times T}$ with $T < S$ such that training the Monte Carlo loss $\mathcal{L}_{\text{aux}}^{\text{MC}}$ results in a representation spanning the top- d left singular vectors of the successor representation.

Denote B_T the top T right singular vectors of the SR. Then the set $\mathcal{G}(B_T)$ satisfies the requirement.

In particular, this finding is consistent in the case where $S = T$ because $\mathcal{G}(B_S) = \{G' \in \mathbb{R}^{S \times T} | \exists M \in \mathcal{O}(T), G' = B_S M\} = \mathcal{G}(I_S)$.

Indeed, let $G' \in \mathcal{G}(B_S)$. There exists $M \in \mathcal{O}(S)$ such that $G' = B_S M = I_S(B_S M)$. Because $B_S M \in \mathcal{O}(S)$, we have $G' \in \mathcal{G}(I_S)$. Hence $\mathcal{G}(B_S) \subset \mathcal{G}(I_S)$.

Let $G' \in \mathcal{G}(I_S)$. There exists $M \in \mathcal{O}(S)$ such that $G' = I_S M = (B_S B_S^\top) M = B_S(B_S^\top M)$. Because $B_S^\top M \in \mathcal{O}(S)$, we have $G' \in \mathcal{G}(B_S)$. Hence $\mathcal{G}(I_S) \subset \mathcal{G}(B_S)$.

As a conclusion, we have $\mathcal{G}(I_S) = \mathcal{G}(B_S)$. \square

Lemma 20. *Let $\{\Phi_\omega\}$ be the set of rotating representations from Figure 6.2 learnt by TD learning with $G = I$ and $d = 2$. All these representations are equally good for learning the main value function by TD learning, that is $\forall \omega \in [0, 1]$,*

$$\mathbb{E}_{\|r\|_2^2 < 1} \left\| \Phi_\omega w_{\Phi_\omega}^{\text{TD}} - V^\pi \right\|_F^2$$

is constant and independent of ω .

Proof. Let's start by considering the case of the three-state circular example. We consider an orthogonal basis for the invariant subspaces of Φ . By definition, $P^\pi e_1 = e_1$, $P^\pi [e_2, e_3] = [e_2, e_3]\Lambda$ so $Le_1 = (1 - \gamma)e_1$ and $L[e_2, e_3] = (I - \gamma P)[e_2, e_3] = [e_2, e_3] - \gamma[e_2, e_3]\Lambda = [e_2, e_3](I - \gamma\Lambda)$.

Assume that there exists $\omega \in [0, 1]$ such that the representation is $\Phi = [e_1, \omega e_2 + (1 - \omega)e_3] = [e_1, e_2, e_3]\Omega$ with $\Omega = \begin{bmatrix} 1 & 0 \\ 0 & \omega \\ 0 & (1 - \omega) \end{bmatrix}$. $L\Phi = [(1 - \gamma)e_1, [e_2, e_3](I - \gamma\Lambda)]\Omega$.

Hence, we have $L\Phi = [e_1, e_2, e_3] \begin{bmatrix} 1 - \gamma & 0 \\ 0 & I - \gamma\Lambda \end{bmatrix} \Omega$

and $\Phi^\top L\Phi = \Omega^\top [e_1, e_2, e_3]^\top [e_1, e_2, e_3] \begin{bmatrix} 1 - \gamma & 0 \\ 0 & I - \gamma\Lambda \end{bmatrix} \Omega = \Omega^\top \begin{bmatrix} 1 - \gamma & 0 \\ 0 & I - \gamma\Lambda \end{bmatrix} \Omega$.

Hence, $(\Phi^\top L\Phi)^{-1} = \begin{bmatrix} (1 - \gamma)^{-1} & 0 \\ 0 & (u^\top (I - \gamma\Lambda) u)^{-1} \end{bmatrix}$ with $u = (w, (1 - w))^\top$. Note that $u^\top (I - \gamma\Lambda) u = \omega^2 \lambda_{1,1} + (1 - \omega)^2 \lambda_{1,1}$

The TD value function is given by $\hat{V}^{\text{TD}} = \Phi(\Phi^\top L\Phi)^{-1}\Phi^\top$

$$\begin{aligned} \hat{V}^{\text{TD}} &= [e_1, e_2, e_3]\Omega \begin{bmatrix} (1 - \gamma)^{-1} & 0 \\ 0 & (u^\top (I - \gamma\Lambda) u)^{-1} \end{bmatrix} \Omega^\top [e_1, e_2, e_3]^\top \\ &= [e_1, e_2, e_3] \begin{bmatrix} (1 - \gamma)^{-1} & 0 \\ 0 & u(u^\top (I - \gamma\Lambda) u)^{-1} u^\top \end{bmatrix} [e_1, e_2, e_3]^\top \\ &= \frac{1/(1 - \gamma)e_1 e_1^\top + \omega^2 e_2 e_2^\top + \omega(1 - \omega)e_3 e_2^\top + \omega(1 - \omega)e_2 e_3^\top + (1 - \omega)^2 e_3 e_3^\top}{\omega^2 \lambda_{1,1} + (1 - \omega)^2 \lambda_{1,1}} \end{aligned}$$

Now $\|\Phi(\Phi^\top L\Phi)^{-1}\Phi^\top - V^\pi\|_F^2$ is independent of ω . □

Lemma 21. *The set of cumulant matrices $G \in \mathbb{R}^{S \times T}$ that preserve the top- T invariant subspaces of the successor representation by TD learning are the top- T orthogonal invariant subspaces of $(I - \gamma P^\pi)^{-1}$, that is satisfying $G^\top G = I$ by orthogonality and $(I - \gamma P^\pi)^{-1}G \subseteq G$ by the invariance property.*

Proof. Let $\Phi \in \mathbb{R}^{S \times d}$ spanning an invariant subspace of L^{-1} . By definition, there exists a block diagonal matrix $J_\Phi \in \mathbb{R}^{d \times d}$ such that $L^{-1}\Phi = \Phi J_\Phi$. Let $G \in O(S, T)$ spanning the top T invariant subspaces of L^{-1} . By definition, there exists a block diagonal matrix $J_G \in \mathbb{R}^{d \times d}$ such that $L^{-1}G = GJ_G$. Hencer, we have

$$\begin{aligned} (L^{-1}GG^\top)\Phi &= (L^{-1}G)G^\top\Phi \\ &= GJ_TG^\top\Phi \\ &= (\Phi J_\Phi) \text{ by orthonormality} \end{aligned}$$

Then, Φ is an invariant subspace of $L^{-1}GG^\top$. □

6.F Proofs for Subsection 6.4.2

We now proceed to the proof of Proposition 4. Before that, we introduce some necessary notations and lemmas.

6.F.1 Notations

Let $O(S, d) := \{A \in \mathbb{R}^{S \times d} : A^\top A = I\}$.

Definition 14. Let $A, B \in O(S, d)$. The principle angles Θ between A and B are given by writing the SVD of $A^\top B = U \cos \Theta V^\top$.

Definition 15. Let $A, B \in O(S, d)$ with principle angles Θ . We define the distance $d(A, B)$ as $d(A, B) := \|\sin \Theta\|_{\text{op}}$.

Proposition 5. Let $A, B \in O(S, d)$. We have the following identities:

$$d(A, B) = \|AA^\top - BB^\top\|_{\text{op}} = \|\sin \Theta\|_{\text{op}} = \|A^\top \bar{B}\|_{\text{op}},$$

where $\bar{B} \in O(S, S - d)$ satisfies $BB^\top + \bar{B}\bar{B}^\top = I$.

6.F.2 Approximate Matrix Decompositions

Lemma 24 (Deterministic error bound). *Let A be an $S \times S$ matrix. Fix $d \leq S$, and partition the SVD of A as:*

$$A = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^\top \\ V_2^\top \end{bmatrix},$$

where Σ_1 is $d \times d$ (the dimensions of all the other factors are determined by this selection). Put $A_d := U_1 \Sigma_1 V_1^\top$ as the rank- d approximation of A . Let Ω be an $S \times \ell$ test matrix ($\ell \geq d$). Put $Y = A\Omega$, $\Omega_1 = V_1^\top \Omega$ and $\Omega_2 = V_2^\top \Omega$. We have that:

$$\|(I - P_Y)A\|_{\text{op}}^2 \leq \|\Sigma_2 \Omega_2 \Omega_1^\dagger\|_{\text{op}}^2.$$

Proof. This proof is adapted from Theorem 9.1 of Halko et al. [2011].

Write $A_d = \hat{U} \hat{\Sigma} \hat{V}^\top$ the full SVD of A_d . By invariance of the spectral norm to unitary transformations,

$$\|(I - P_Y)A_d\|_{\text{op}}^2 = \|\hat{U}^\top (I - P_Y) \hat{U} (\hat{U}^\top A_d)\|_{\text{op}}^2 = \|(I - P_{\hat{U}^\top Y}) (\hat{U}^\top A_d)\|_{\text{op}}^2$$

Assume the diagonal entries of Σ_2 are not all strictly positive. Then Σ_2 is zero as a consequence of the ordering of the singular values.

$$\text{range}(\hat{U}^\top Y) = \text{range} \begin{bmatrix} \Sigma_1 \Omega_1 \\ 0 \end{bmatrix} = \text{range} \begin{bmatrix} \Sigma_1 V_1^\top \\ 0 \end{bmatrix} = \text{range}(\hat{U}^\top A_d)$$

So we can conclude that $\|(I - P_Y)A_d\|_{\text{op}}^2 = 0$ assuming that V_1^\top and Ω_1 have full row rank.

Now assume that the diagonal entries of Σ_1 are strictly positive. Let $Z = \hat{U}^\top Y \cdot \Omega_1^\dagger \Sigma_1^{-1} = \begin{bmatrix} I_d \\ F \end{bmatrix}$ with $F = \Sigma_2 \Omega_2 \Omega_1^\dagger \Sigma_1^{-1} \in \mathbb{R}^{(S-d) \times d}$.

By construction, $\text{range}(Z) \subset \text{range}(\hat{U}^\top Y)$, hence we have,

$$\begin{aligned} \|(I - P_{\hat{U}^\top Y})(\hat{U}^\top A_d)\|_{\text{op}}^2 &\leq \|(I - P_Z) \hat{U}^\top A_d\|_{\text{op}}^2 \\ &\leq \|A_d^\top \hat{U} (I - P_Z) \hat{U}^\top A_d\|_{\text{op}} \\ &\leq \|\hat{\Sigma} (I - P_Z) \hat{\Sigma}\|_{\text{op}} \end{aligned}$$

Following the proof from Theorem 9.1 of Halko et al. [2011], we have

$$(I - P_Z) \preceq \begin{bmatrix} F^\top F & B \\ B^\top & I_{S-d} \end{bmatrix}$$

where $B = -(I_d - F^\top F)^{-1} F^\top \in \mathbb{R}^{d \times (S-d)}$.

Consequently, we have

$$\hat{\Sigma}(I - P_Z)\hat{\Sigma} \preceq \begin{bmatrix} \Sigma_1 F^\top F \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix}$$

$\hat{\Sigma}(I - P_Z)\hat{\Sigma}$ is PSD by the conjugation rule, hence the matrix on the right hand side is PSD too. It follows that

$$\|\hat{\Sigma}(I - P_Z)\hat{\Sigma}\|_{\text{op}} \leq \|\Sigma_1 F^\top F \Sigma_1\|_{\text{op}} = \|F \Sigma_1\|_{\text{op}}^2 = \|\Sigma_2 \Omega_2 \Omega_1^\dagger\|_{\text{op}}^2$$

□

Lemma 25 (Average spectral error). *Let A be an $S \times S$ matrix with singular values $\sigma_1 \geq \sigma_2 \geq \dots$. Fix a target rank $2 \leq d \leq S$ and an oversampling parameter $p \geq 2$ where $p + d \geq S$. Draw and $S \times (d + p)$ standard gaussian matrix Ω and construct the sample matrix $Y = A\Omega$. Then, we have*

$$\mathbb{E}\|(I - P_Y)A_d\|_{\text{op}} \leq \sqrt{\frac{d}{p-1}}\sigma_{d+1} + \frac{e\sqrt{d+p}}{p} \left(\sum_{j=d+1}^S \sigma_j^2 \right)^{1/2}.$$

Proof. By Lemma 24 and linearity of the expectation, we have

$$\begin{aligned} \mathbb{E}\|(I - P_Y)A_d\|_{\text{op}} &\leq \mathbb{E}\|\Sigma_2 \Omega_2 \Omega_1^\dagger\|_{\text{op}} \\ &\leq \sqrt{\frac{d}{p-1}}\sigma_{d+1} + \frac{e\sqrt{d+p}}{p} \left(\sum_{j=d+1}^S \sigma_j^2 \right)^{1/2}, \end{aligned}$$

where the last inequality comes from Theorem 10.6 of Halko et al. [2011]. □

Lemma 26. *Let $A \in \mathbb{R}^{m \times n}$, and fix a $d < n$. Let $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$ denote the singular values of M listed in decreasing order, and suppose that $\sigma_k > 0$. Let A_d denote the rank- d approximation of A . Fix any matrix $Y \in \mathbb{R}^{m \times T}$. We have:*

$$\|(I - P_Y)A_k\|_{\text{op}} \geq \|(I - P_Y)P_{A_k}\|_{\text{op}}\sigma_k.$$

Proof. Decompose $P_Y^\perp A_k$ as:

$$P_Y^\perp A_k = P_Y^\perp P_{A_k} A_k$$

$$\|P_Y^\perp A_k\|_{\text{op}} = \|P_Y^\perp P_{A_k} A_k\|_{\text{op}} \geq \|P_Y^\perp P_{A_k}\|_{\text{op}} \|A_k\|_{\text{op}} = \|P_Y^\perp P_{A_k}\|_{\text{op}} \sigma_k$$

where the inequality comes from the sub-multiplicativity of the the operator norm \square

Proposition 6. *Let A be an $S \times S$ matrix with singular values $\sigma_1 \geq \sigma_2 \geq \dots$. Fix a target rank $2 \leq d \leq n$ and an oversampling parameter $p \geq 2$ where $p + d \geq S$. Draw an $n \times (d + p)$ standard gaussian matrix Ω and construct the sample matrix $Y = A\Omega$. Then, we have*

$$\mathbb{E}\|(I - P_Y)P_{A_d}\|_{\text{op}} \leq \sqrt{\frac{d}{p-1} \frac{\sigma_{d+1}}{\sigma_d}} + \frac{e\sqrt{d+p}}{p} \left(\sum_{j=d+1}^S \frac{\sigma_j^2}{\sigma_d^2} \right)^{1/2}.$$

Proof. By Lemma 26 and linearity of the expectation, we have

$$\frac{1}{\sigma_d} \mathbb{E}\|(I - P_Y)A_d\|_{\text{op}} \geq \mathbb{E}\|(I - P_Y)P_{A_d}\|_{\text{op}}$$

Now applying Lemma 25, we have

$$\sqrt{\frac{d}{p-1} \frac{\sigma_{d+1}}{\sigma_d}} + \frac{e\sqrt{d+p}}{p} \left(\sum_{j=d+1}^S \frac{\sigma_j^2}{\sigma_d^2} \right)^{1/2} \geq \mathbb{E}\|(I - P_Y)P_{A_d}\|_{\text{op}}$$

\square

Observe that, as the oversampling factor p grows, the RHS tends to zero. However, the dependence will be something like $p \gtrsim 1/\varepsilon^2$, if you want the RHS to be $\leq \varepsilon$. This actually makes sense I think– you are using concentration of measure to increase the accuracy, so you should pay $1/\varepsilon^2$ sample complexity.

6.F.3 Analysis

Proposition 4 (MC Error bound). *Let $G \in \mathbb{R}^{S \times T}$ be a sample from a standard gaussian distribution and assume $d \leq T$. Let F_d be the top- d left singular vectors of the successor representation $(I - \gamma P^\pi)^{-1}$ and \hat{F}_d be the top left singular vectors of $(I - \gamma P^\pi)^{-1}G$. Denote $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_S$ the singular values of the SR and $\text{dist}(F_d, \hat{F}_d)$ the sin θ distance between the subspaces spanned by F_d and \hat{F}_d . We have*

$$\mathbb{E}[\text{dist}(F_d, \hat{F}_d)] \leq \sqrt{\frac{d}{T-d-1} \frac{\sigma_{d+1}}{\sigma_d}} + \frac{e\sqrt{T}}{T-d} \left(\sum_{j=d+1}^n \frac{\sigma_j^2}{\sigma_d^2} \right)^{\frac{1}{2}}$$

Proof. Let $l \in \{d, \dots, S\}$. $F_l \in O(S, l)$ be the top l left singular vectors of $(I - \gamma P^\pi)^{-1}$ and $\hat{F}_l \in O(S, d)$ be the top left singular vectors of $(I - \gamma P^\pi)^{-1}G$.

$$\begin{aligned}
d(F_d, \hat{F}_d) &= \|\hat{F}_d^\top F_d^\perp\|_{\text{op}} \\
&= \|P_{\hat{F}_d} P_{F_d}^\perp\|_{\text{op}} \\
&\leq \|P_{L^{-1}G} P_{F_d}^\perp\|_{\text{op}} \text{ as } \text{span}(\hat{F}_d) \subseteq \text{span}(L^{-1}G) \\
&= \|\hat{F}_T^\top F_d^\perp\|_{\text{op}} \\
&= \|F_d^\top \hat{F}_T^\perp\|_{\text{op}} \\
&= \|P_{F_d} P_{\hat{F}_T}^\perp\|_{\text{op}} \\
&= \|P_{\hat{F}_T}^\perp P_{F_d}\|_{\text{op}} \text{ by symmetry of the projection matrices} \\
&= \|(I - P_{\hat{F}_T})P_{F_d}\|_{\text{op}} \\
&= \|(I - P_{L^{-1}G})P_{(L^{-1})_d}\|_{\text{op}} \\
&\leq \frac{1}{\sigma_d} \|(I - P_{L^{-1}G})(L^{-1})_d\|_{\text{op}} \text{ by Lemma 26}
\end{aligned}$$

Now taking the expectation with respect to G and applying Proposition 6,

$$\mathbb{E}[d(F_d, \hat{F}_d)] \leq \sqrt{\frac{d}{T-d-1}} \frac{\sigma_{d+1}}{\sigma_d} + \frac{e\sqrt{T}}{T-d} \left(\sum_{j=d+1}^n \frac{\sigma_j^2}{\sigma_d^2} \right)^{1/2}.$$

□

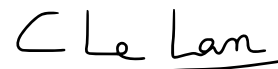
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor (**only required where there isn't already a statement of contribution within the paper itself**).

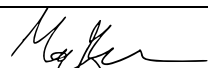
Title of Paper	Bootstrapped Representations in Reinforcement Learning
Publication Status	<input checked="" type="checkbox"/> Published <input type="checkbox"/> Accepted for Publication <input type="checkbox"/> Submitted for Publication <input type="checkbox"/> Unpublished and unsubmitted work written in a manuscript style
Publication Details	Charline Le Lan, Stephen Tu, Mark Rowland, Anna Harutyunyan, Rishabh Agarwal, Marc G. Bellemare, Will Dabney. <i>In Reincarnating RL Workshop at ICLR 2023.</i>

Student Confirmation

Student Name:	Charline Le Lan		
Contribution to the Paper	<p>I led the project, wrote the paper, came up with an initial proof about the representations learnt by TD as a motivating step for this project, wrote proofs in appendix, wrote the code for synthetic matrices experiments and random cumulants on the four-room domain.</p> <p>Will suggested the cumulant approach based on CPC with clustering, implemented this method and generated the plots for the offline pre-training experiments.</p> <p>All authors contributed to discussions, advising and provided feedback and edits on the manuscript.</p>		
Signature		Date	March 23, 2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Marc G. Bellemare			
Supervisor comments	Following our prior work on representation learning, Charline discovered an important gap in our understanding of how deep reinforcement learning algorithms actually operate, because they bootstrap rather than use supervised labels. This work is not yet formally published but has received glowing reviews at ICML23.		
Signature		Date	13/04/23

This completed form should be included in the thesis, at the end of the relevant chapter.

7

Discussion

To conclude, we summarize the main contributions presented in this dissertation and discuss promising avenues for future work.

7.1 Conclusion

The thesis defended in this dissertation is that topological tools, statistical learning and dynamical systems theory can help shape principled state representations that enhance RL agents.

We demonstrated this thesis by considering state representations through two approaches, state abstraction (Chapter 3) and state features (Chapter 4, Chapter 5, Chapter 6), and introduced a framework to improve reinforcement learning algorithms in a theoretically-grounded manner.

In Chapter 3, we studied the goodness of a state-aggregation depending on how well it can extrapolate values or q-values. This is key for algorithms like approximate value iteration, policy iteration and for exploration algorithms on continuous state spaces [Pazis and Parr, 2013]. This work also informs the choice of a behavioral metric and corroborates the empirical success of follow-up state-similarity based algorithms [Castro et al., 2021, Zhang et al., 2020].

In Chapter 4, we compared the generalization capacity of different state representations by looking at their effective dimension, a quantity we introduced and which drives the generalization to unseen states. Our analysis also motivated a new auxiliary learning rule aiming at improving generalization of deep RL agents and our work inspired several empirical investigations, in particular in the offline setting [Fu et al., 2022].

In Chapter 5, we provided an algorithm to scale Monte Carlo auxiliary tasks without increasing the amount of memory space required all while recovering the desired state representation. We believe this approach will be crucial to scale up auxiliary tasks such as proto-value networks [Farebrother et al., 2023].

In Chapter 6, we set up a mathematical model with which we characterize the representations learned when predicting a collection of auxiliary tasks. This analysis is important because, given a desired state representation, it informs which additional predictions the agent should make and how it should make these predictions. Together with this analysis, we developed scalable algorithms for sparse reward settings and offline pre-training which proved useful on the Four Rooms and Mountain Car domains.

7.2 Future Directions

There are a number of exciting directions for future work following from this dissertation. We discuss a few of them in the following sections.

7.2.1 Further Theoretical Analysis of Representation Learning Schemes

A direct continuation of the work presented in this thesis would be to extend our work from Chapter 4 by deriving a generalization bound from a given representation in the temporal difference setting. A promising way would be to use concentration inequalities and arguments for the analysis of random design linear least-squares problems [Hsu et al., 2014]. We believe that the effective dimension from Chapter 4 would also play an important role. Recently, Duan et al. [2021] analyzed the

estimation error when learning by kernel least-squares temporal difference (LSTD) under a generative model. However, their proof is more complex than the one suggested above and Duan et al. [2021] do not aim at comparing state representations under the lens of generalization. A generalization bound in the TD setting would be beneficial because it could lead to novel auxiliary update rules in the future.

We believe it is also essential to develop a better theoretical understanding of how different representation learning schemes relate to each other.

Recent approaches about representation learning in RL are generally categorized into three different families of methods: auxiliary tasks on which we focus in this thesis, contrastive [Chen et al., 2020a, He et al., 2020, Chen et al., 2020b] and non contrastive self-supervised approaches [Grill et al., 2020, Chen and He, 2021]. Usually, practitioners tend to think of them separately.

Lately, HaoChen et al. [2021] analysed contrastive learning in the setting of classification and showed a close relationship with spectral decomposition. We demonstrated in Chapter 6 that training auxiliary tasks in a supervised way also performs a spectral decomposition on the auxiliary task matrix. It would thus be interesting to characterize to which extend the representations learned with contrastive losses (e.g. by means of a latent embedding) are connected with the ones learned with auxiliary tasks.

Recently, Garrido et al. [2023] showed that contrastive and non-contrastive methods are to some extent theoretically equivalent. Following their theoretical insights, they demonstrated that careful design choices could close the gap between the two approaches empirically.

Looking forward, we believe that extending some of these theoretical results to the RL setting would be an exciting path for future research. Understanding similarities and potential differences between the representations resulting from training contrastive, self-supervised and auxiliary tasks methods could result in the development of more principled representation learning algorithms for RL.

7.2.2 Benchmarks

In addition to the theoretical comparison suggested in Subsection 7.2.1, a complementary approach would be to develop a unified benchmark of different representation learning approaches.

In this dissertation, we relied on the Atari 2600 video games [Bellemare et al., 2013] and smaller domains such as the Four Rooms domain [Sutton et al., 1999], Mountain Car [Moore, 1990] or Puddle World [Sutton, 1995] to demonstrate that our theoretical insights made useful predictions about value-based deep RL agents (Chapter 4 and Chapter 6) and led to efficient novel algorithms (Chapter 4, Chapter 5, Chapter 6). Several approaches also rely on spectral representations [Barreto et al., 2017a,b, Janner et al., 2020, Blier et al., 2021, Touati and Ollivier, 2021, Farebrother et al., 2023, Schwarzer et al., 2021, Guo et al., 2020] but have often been evaluated on different benchmarks and regimes.

Moving forward, an interesting direction would be to carefully tune the hyperparameters and the network architectures of these methods on the same benchmarks, in line with the possible equivalences suggested in Subsection 7.2.1. Recently, Touati et al. [2023] started to compare different representation learning algorithms on more tasks and environments from the Unsupervised RL benchmark [Laskin et al., 2021] but their experiments are still small-scale and the environments deterministic. It would be exciting to evaluate these methods on other domains, for instance focusing on complex or sparse reward environments. The NetHack learning environment [Küttler et al., 2020] is a large-scale stochastic domain which would be an interesting evaluation platform for the auxiliary tasks methods presented in Chapter 6. The video game Minecraft could also be used as a testbed. It is a challenging benchmark due to the high dimensionality of its state and action space as well as the sparsity of its reward signal. The Balloon Learning Environment [BLE; Greaves et al., 2021] is a partially observable and non-stationary environment which simulates the real-world problem of navigating stratospheric balloons [Bellemare et al., 2020]. It would be useful to evaluate the sample and compute-efficiency of the various representation learning algorithms we mentioned on this task. Finally, as a way to measure the

goodness of these representation learning methods for performing actions in the real world, we could rely on some robotics benchmarks, for instance the robotic block stacking environment [Lee et al., 2021]. Two settings would be particularly of interest: 1) first pretraining representations from offline data and then using them to learn a policy in an online phase [Farebrother et al., 2023] and 2) pretraining representations from offline data and fine-tuning them during second offline training phase.

Applying the methods presented in this thesis to these benchmarks would require scaling them up. Farebrother et al. [2023] suggested that, given fixed capacity, the performance of value-based agents saturated as the number of auxiliary tasks used to pre-train representations kept increasing. Fully scaling up auxiliary-task based methods would require revisiting some of the architectural choices made so far. For instance, modern architecture, more depth, adaptive width and combining features in a non-linear way could provide a path to keep scaling up these methods. We also demonstrated theoretically in Chapter 6 and empirically in Chapter 5 that learning dynamics differ when the weights of the last linear layer of a deep RL architecture are parameterized explicitly or implicitly, as a function of the state features. Comparing both training dynamics on the above benchmarks would also be a way to work towards scaling up auxiliary tasks in reinforcement learning.

In addition, we presented in this thesis several quantities to measure the usefulness of a representation such as the approximation error, the generalization error (Chapter 4), the computational cost Chapter 3, an l_1 -ball optimality criterion (Chapter 6) and the stability [Ghosh and Bellemare, 2020]. Reporting these quantities for different representation learning methods would be an additional helpful tool to evaluate them.

7.2.3 Pre-training Representations and Reincarnating Reinforcement Learning

A thrilling direction we touched upon in Chapter 6 is the idea of pre-training representations. It is part of an emerging trend of reusing computation in RL, also referred to as reincarnating RL [Agarwal et al., 2022]. Leveraging prior

computation is exciting because it can speed up training which is necessary to develop large scale RL systems. In particular, reusing computation in the forms of pre-trained representation can help learning a control policy faster than learning it tabula rasa. These representations are often learned from offline datasets of transitions as part of an unsupervised pre-training phase [Touati and Ollivier, 2021, Farebrother et al., 2023, Touati et al., 2023]. They can be fixed and used for linear function approximation [Farebrother et al., 2023] or fine-tuned throughout training (see e.g. Subsection 6.5.3).

In this thesis, we looked at designing algorithms in single-environment settings. However, learning more general agents capable of solving several tasks (e.g. games) could further increase their generalization capacities. A possible research goal is to develop universal pre-training methods of reinforcement learning agents [Guo et al., 2020, Chen et al., 2021, Reed et al., 2022, Venuto et al., 2022, Taiga et al., 2023]. We could therefore investigate the goodness of some of the auxiliary-task based methods introduced in this thesis at learning universal features, for instance on Atari video games. Following Agarwal et al. [2022], open-sourcing these features would enable the research community to tackle RL problems requiring significant computational resources by focusing on credit assignment separately from representation learning.

There are several promising real-world applications to reusing pre-trained representations. Specifically, some of the work in this thesis could be leveraged for dialogue systems. Current conversational response models such as ChatGPT [OpenAI, 2023] are trained to predict the next text token, which simply corresponds to behavior cloning [Pomerleau, 1991, Bagnell et al., 2006, Ross and Bagnell, 2010]. However, the Internet can be seen as an environment with non-trivial dynamics providing high quantity but low quality data and Schwarzer et al. [2021], Farebrother et al. [2023] demonstrated that self-supervised and auxiliary losses outperformed behavioral cloning on "suboptimal" Atari data. Hence, investigating alternative ways to pre-train large language models by leveraging the successor representation is an exciting challenge ahead of us.

Bibliography

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Zheng Xiaoqiang. Tensorflow: A system for large-scale machine learning. In *USENIX Symposium on Operating Systems Design and Implementation*, 2016.

David Abel, David Hershkowitz, and Michael Littman. Near optimal behavior via approximate state abstraction. In *Proceedings of the International Conference on Machine Learning*, 2016.

Naman Agarwal, Brian Bullins, and Elad Hazan. Second-order stochastic optimization in linear time. *Journal of Machine Learning Research*, 18:1–40, 2017.

Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2020.

Rishabh Agarwal, Marlos C. Machado, Pablo Samuel Castro, and Marc G Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In *Proceedings of the International Conference on Learning Representations*, 2021a.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, 2021b.

- Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Reincarnating reinforcement learning: Reusing prior computation to accelerate progress. In *Advances in Neural Information Processing Systems*, 2022.
- Ehsan Amid and Manfred K Warmuth. An implicit form of krasulina’s k-PCA update without the orthonormality constraint. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.
- Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, 2017.
- T. W. Archibald, K. I. M. McKinnon, and L. C. Thomas. On the generation of Markov decision processes. *The Journal of the Operational Research Society*, 46(3):354–361, 1995.
- J Bagnell, Joel Chestnutt, David Bradley, and Nathan Ratliff. Boosting structured prediction for imitation learning. In *Advances in Neural Information Processing Systems*, 2006.
- Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995.
- Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.
- Laura Balzano. On the equivalence of oja’s algorithm and grouse. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2022.
- Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. In *Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2010.

- Etienne Barnard. Temporal-difference methods and markov models. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(2):357–365, 1993.
- André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017a.
- André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017b.
- Bahram Behzadian and Marek Petrik. Low-rank feature selection for reinforcement learning. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics*, 2018.
- Marc Bellemare, Will Dabney, Robert Dadashi, Adrien Ali Taiga, Pablo Samuel Castro, Nicolas Le Roux, Dale Schuurmans, Tor Lattimore, and Clare Lyle. A geometric perspective on optimal representations for reinforcement learning. In *Advances in Neural Information Processing Systems*, 2019.
- Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.
- Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2017.
- Marc G Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Marlos C Machado, Subhodeep Moitra, Sameera S Ponda, and Ziyu Wang. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836):77–82, 2020.
- Marc G. Bellemare, Will Dabney, and Mark Rowland. *Distributional Reinforcement Learning*. MIT Press, 2023. <http://www.distributional-rl.org>.

- RJNJ Bellman. Dynamic programming princeton university press princeton. *New Jersey Google Scholar*, 1957.
- Dimitri P. Bertsekas. Approximate policy iteration: A survey and some new methods. *Journal of Control Theory and Applications*, 9(3):310–335, 2011.
- Léonard Blier, Corentin Tallec, and Yann Ollivier. Learning successor states and goal-dependent values: A mathematical viewpoint. *arXiv preprint arXiv:2101.07123*, 2021.
- Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.
- Justin A Boyan. Technical update: Least-squares temporal difference learning. *Machine learning*, 49(2):233–246, 2002.
- James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL <http://github.com/google/jax>.
- Steven J Bradtke and Andrew G Barto. Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1):33–57, 1996.
- Pierre Brémaud. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*, volume 31. Springer Science & Business Media, 2013.
- Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *Proceedings of the International Conference on Learning Representations*, 2018.
- Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

- Pablo S. Castro, Subhodeep Moitra, Carles Gelada, Saurabh Kumar, and Marc G. Bellemare. Dopamine: A research framework for deep reinforcement learning. *arXiv*, 2018.
- Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.
- Pablo Samuel Castro and Doina Precup. Using bisimulation for policy transfer in mdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2010.
- Pablo Samuel Castro, Prakash Panangaden, and Doina Precup. Notions of state equivalence under partial observability. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2009.
- Pablo Samuel Castro, Tyler Kastner, Prakash Panangaden, and Mark Rowland. Mico: Improved representations via sampling-based state similarity for markov decision processes. In *Advances in Neural Information Processing Systems*, 2021.
- Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, 2021.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning*, 2020a.
- Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. In *Advances in Neural Information Processing Systems*, 2020b.
- Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021.

- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 2017.
- Wesley Chung, Somjit Nath, Ajin Joseph, and Martha White. Two-timescale networks for nonlinear value function approximation. In *Proceedings of the International Conference on Learning Representations*, 2018.
- Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2018a.
- Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018b.
- Will Dabney, André Barreto, Mark Rowland, Robert Dadashi, John Quan, Marc G Bellemare, and David Silver. The value-improvement path: Towards better representations for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.
- Robert Dadashi, Adrien Ali Taiga, Nicolas Le Roux, Dale Schuurmans, and Marc G Bellemare. The value function polytope in reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2019.
- Wei Dai and Olgica Milenkovic. Set: An algorithm for consistent matrix completion. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010.
- Christoph Dann, Gerhard Neumann, Jan Peters, et al. Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Research*, 15:809–883, 2014.
- John M Danskin. *The theory of max-min and its application to weapons allocation problems*, volume 5. Springer Science & Business Media, 2012.

- Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993.
- Daniela Pucci De Farias. *The linear programming approach to approximate dynamic programming*. John Wiley & Sons, 2003.
- Christopher De Sa, Christopher Re, and Kunle Olukotun. Global convergence of stochastic gradient descent for some non-convex matrix problems. In *Proceedings of the International Conference on Machine Learning*, 2015.
- Zhijie Deng, Jiaxin Shi, and Jun Zhu. Neuraief: Deconstructing kernels by deep neural networks. In *Proceedings of the International Conference on Machine Learning*, 2022.
- Yaqi Duan, Chi Jin, and Zhiyuan Li. Risk bounds and rademacher complexity in batch reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2021.
- Jesse Farebrother, Joshua Greaves, Rishabh Agarwal, Charline Le Lan, Ross Goroshin, Pablo Samuel Castro, and Marc G Bellemare. Proto-value networks: Scaling representation learning with auxiliary tasks. In *Proceedings of the International Conference on Learning Representations*, 2023.
- Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatin, Alexander Novikov, Francisco J. R. Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, David Silver, Demis Hassabis, and Pushmeet Kohli. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022. doi: 10.1038/s41586-022-05172-4.
- William Fedus, Carles Gelada, Yoshua Bengio, Marc G Bellemare, and Hugo Larochelle. Hyperbolic discounting and learning over multiple horizons. *arXiv preprint arXiv:1902.06865*, 2019.

- Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2004.
- Norman Ferns, Prakash Panangaden, and Doina Precup. Metrics for Markov Decision Processes with infinite state spaces. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2005.
- Vincent François-Lavet, Yoshua Bengio, Doina Precup, and Joelle Pineau. Combined reinforcement learning via abstract representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.
- Yuwei Fu, Di Wu, and Benoit Boulet. A closer look at offline rl agents. In *Advances in Neural Information Processing Systems*, 2022.
- Quentin Garrido, Yubei Chen, Adrien Bardes, Laurent Najman, and Yann Lecun. On the duality between contrastive and non-contrastive self-supervised learning. In *Proceedings of the International Conference on Learning Representations*, 2023.
- Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Proceedings of the Conference on Learning Theory*, 2015.
- Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *Proceedings of the International Conference on Machine Learning*, 2017.
- Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. DeepMDP: Learning continuous latent space models for representation learning. In *Proceedings of the International Conference on Machine Learning*, 2019.
- Ian Gemp, Brian McWilliams, Claire Vernade, and Thore Graepel. EigenGame: Pca as a nash equilibrium. In *Proceedings of the International Conference on Learning Representations*, 2021.

- Ian Gemp, Brian McWilliams, Claire Vernade, and Thore Graepel. EigenGame unloaded: When playing games is better than optimizing. In *Proceedings of the International Conference on Learning Representations*, 2022.
- Dibya Ghosh and Marc G Bellemare. Representations for stable off-policy reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2020.
- Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in markov decision processes. *Artificial Intelligence*, 147(1-2):163–223, 2003.
- Israel Gohberg, Peter Lancaster, and Leiba Rodman. *Invariant subspaces of matrices with applications*. SIAM, 2006.
- Robert M Gray et al. Toeplitz and circulant matrices: A review. *Foundations and Trends® in Communications and Information Theory*, 2(3):155–239, 2006.
- Joshua Greaves, Salvatore Candido, Vincent Dumoulin, Ross Goroshin, Sameera S. Ponda, Marc G. Bellemare, and Pablo Samuel Castro. Balloon Learning Environment, 2021. URL <https://github.com/google/balloon-learning-environment>.
- Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent—a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020.
- Charles Miller Grinstead and James Laurie Snell. *Introduction to probability*. American Mathematical Soc., 2012.
- Carlos Guestrin, Daphne Koller, Ronald Parr, and Shobha Venkataraman. Efficient solution algorithms for factored mdps. *Journal of Artificial Intelligence Research*, 19:399–468, 2003.

- Caglar Gulcehre, Ziyu Wang, Alexander Novikov, Tom Le Paine, Sergio Gómez Colmenarejo, Konrad Zolna, Rishabh Agarwal, Josh Merel, Daniel Mankowitz, Cosmin Paduraru, et al. RL unplugged: a suite of benchmarks for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.
- Zhaohan Daniel Guo, Bernardo Avila Pires, Bilal Piot, Jean-Bastien Grill, Florent Alché, Rémi Munos, and Mohammad Gheshlaghi Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2020.
- Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.
- Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. In *Advances in Neural Information Processing Systems*, 2021.
- Moritz Hardt. Understanding alternating minimization for matrix completion. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 651–660. IEEE, 2014.
- Charles R Harris, K Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825): 357–362, 2020.
- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.
- Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver.

- Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.
- Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.
- Daniel Hsu, Sham Kakade, and Tong Zhang. A tail inequality for quadratic forms of subgaussian random vectors. *Electronic Communications in Probability*, 17: 1–6, 2012.
- Daniel Hsu, Sham M. Kakade, and Tong Zhang. Random design analysis of ridge regression. *Foundations of Computational Mathematics*, 14(3):569–600, 2014.
- John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95, 2007.
- Hutchinson, Le Lan, Zaidi, Emilien Dupont, Yee Whye Teh, and Hyunjik Kim. LieTransformer: Equivariant self-attention for lie groups. In *Proceedings of the International Conference on Machine Learning*, 2021.
- Max Jaderberg, Volodymyr Mnih, Wojciech M. Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *Proceedings of the International Conference on Learning Representations*, 2017.
- Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the ACM Symposium on Theory of Computing*, 2013.
- Michael Janner, Igor Mordatch, and Sergey Levine. Gamma-models: Generative temporal difference learning for infinite-horizon prediction. In *Advances in Neural Information Processing Systems*, 2020.
- Chi Jin, Sham M Kakade, and Praneeth Netrapalli. Provable efficient online matrix completion via non-convex stochastic gradient descent. In *Advances in Neural Information Processing Systems*, 2016.

- Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *Proceedings of the International Conference on Machine Learning*, 2017.
- Eric Jones, Travis Oliphant, and Pearu Peterson. SciPy: open source scientific tools for Python, 2001. URL <http://www.scipy.org>.
- Sham Kakade, Michael J Kearns, and John Langford. Exploration in metric state spaces. In *Proceedings of the International Conference on Machine Learning*, 2003.
- Steven Kapturowski, Georg Ostrovski, John Quan, Remi Munos, and Will Dabney. Recurrent experience replay in distributed reinforcement learning. In *Proceedings of the International conference on learning representations*, 2019.
- Philipp W Keller, Shie Mannor, and Doina Precup. Automatic basis function construction for approximate dynamic programming and reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2006.
- John G Kemeny and J Laurie Snell. Finite continuous time markov chains. *Theory of Probability & Its Applications*, 6(1):101–105, 1961.
- Raghunandan H Keshavan and Sewoong Oh. A gradient descent algorithm on the grassman manifold for matrix completion. *arXiv preprint arXiv:0910.5260*, 2009.
- Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015.
- George D. Konidaris, Sarah Osentoski, and Philip S. Thomas. Value function approximation in reinforcement learning using the fourier basis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2011.

- T Krasulina. Method of stochastic approximation in the determination of the largest eigenvalue of the mathematical expectation of random matrices. *Automatation and remote control*, 2:50–56, 1970.
- R Matthew Kretchmar and Charles W Anderson. Using temporal neighborhoods to adapt function approximators in reinforcement learning. In *International Work Conference on Artificial and Natural Neural Networks*, 1999.
- Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. In *Proceedings of the International Conference on Learning Representations*, 2021.
- Aviral Kumar, Rishabh Agarwal, Tengyu Ma, Aaron Courville, George Tucker, and Sergey Levine. Dr3: Value-based deep reinforcement learning requires explicit regularization. In *Proceedings of the International Conference on Learning Representations*, 2022.
- Heinrich Küttler, Nantas Nardelli, Alexander Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. The nethack learning environment. In *Advances in Neural Information Processing Systems*, 2020.
- Branislav Kveton and Milos Hauskrecht. Learning basis functions in hybrid domains. In *Proceedings of the National Conference on Artificial Intelligence*, 2006.
- Michail G Lagoudakis and Ronald Parr. Least-squares policy iteration. *The Journal of Machine Learning Research*, 4:1107–1149, 2003.
- Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. Cic: Contrastive intrinsic control for unsupervised skill discovery. In *Deep RL Workshop at NeurIPS*, 2021.
- Charline Le Lan and Rishabh Agarwal. Revisiting bisimulation: a sampling-based state similarity pseudo-metric. *Preprint*, 2023.

- Charline Le Lan and Laurent Dinh. Perfect density models cannot guarantee anomaly detection. *Entropy*, 23(12):1690, 2021.
- Charline Le Lan, Marc G Bellemare, and Pablo Samuel Castro. Metrics and continuity in reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.
- Charline Le Lan, Stephen Tu, Adam Oberman, Rishabh Agarwal, and Marc G Bellemare. On the generalization of representations in reinforcement learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2022.
- Charline Le Lan, Joshua Greaves, Jesse Farebrother, Mark Rowland, Fabian Pedregosa, Rishabh Agarwal, and Marc G Bellemare. A novel stochastic gradient descent algorithm for learning principal subspaces. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2023a.
- Charline Le Lan, Stephen Tu, Mark Rowland, Anna Harutyunyan, Rishabh Agarwal, Marc G Bellemare, and Will Dabney. Bootstrapped representations in reinforcement learning. In *Workshop on Reincarnating Reinforcement Learning at ICLR*, 2023b.
- Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.
- Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist>, 2, 2010.
- Alex X Lee, Coline Manon Devin, Yuxiang Zhou, Thomas Lampe, Konstantinos Bousmalis, Jost Tobias Springenberg, Arunkumar Byravan, Abbas Abdolmaleki, Nimrod Gileadi, David Khosid, et al. Beyond pick-and-place: Tackling robotic stacking of diverse shapes. In *Proceedings of the Conference on Robot Learning*, 2021.

- Nir Levine, Tom Zahavy, Daniel Mankowitz, Aviv Tamar, and Shie Mannor. Shallow updates for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017.
- Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.
- Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for mdps. In *ISAIM*, 2006.
- Xingguo Li, Junwei Lu, Raman Arora, Jarvis Haupt, Han Liu, Zhaoran Wang, and Tuo Zhao. Symmetry, saddle points, and global optimization landscape of nonconvex matrix factorization. *IEEE Transactions on Information Theory*, 65(6):3489–3514, 2019.
- Guoqing Liu, Chuheng Zhang, Li Zhao, Tao Qin, Jinhua Zhu, Li Jian, Nenghai Yu, and Tie-Yan Liu. Return-based contrastive representation learning for reinforcement learning. In *Proceedings of the International Conference on Learning Representations*, 2021.
- Clare Lyle, Mark Rowland, Georg Ostrovski, and Will Dabney. On the effect of auxiliary tasks on representation dynamics. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2021.
- Marlos C. Machado, Clemens Rosenbaum, Xiaoxiao Guo, Miao Liu, Gerald Tesauro, and Murray Campbell. Eigenoption discovery through the deep successor representation. In *Proceedings of the International Conference on Learning Representations*, 2018.
- M.C. Machado, M.G. Bellemare, and M. Bowling. A Laplacian framework for option discovery in reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2017.

- Sephora Madjiheurem and Laura Toni. Representation learning on graphs: A reinforcement learning application. In *Proceedings of the International Conference on Machine Learning*, 2019.
- Sridhar Mahadevan and Mauro Maggioni. Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *Journal of Machine Learning Research*, 8(10), 2007.
- Odalric-Ambrym Maillard and Rémi Munos. Compressed least-squares regression. In *Advances in Neural Information Processing Systems*, 2009.
- Emile Mathieu, Charline Le Lan, Chris J Maddison, Ryota Tomioka, and Yee Whye Teh. Continuous hierarchical representations with poincaré variational auto-encoders. In *Advances in Neural Information Processing Systems*, 2019.
- Ishai Menache, Shie Mannor, and Nahum Shimkin. Basis function adaptation in temporal difference reinforcement learning. *Annals of Operations Research*, 134(1):215–238, 2005.
- Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nazi, et al. A graph placement methodology for fast chip design. *Nature*, 594(7862):207–212, 2021.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dhharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

- Andrew William Moore. Efficient memory-based learning for robot control. Technical report, University of Cambridge, 1990.
- Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, 2016.
- Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning*, 2010.
- Andriy Norets. Continuity and differentiability of expected value functions in dynamic discrete choice models. *Quantitative economics*, 1(2):305–322, 2010.
- Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.
- Jungseul Ok, Alexandre Proutiere, and Damianos Tranos. Exploration in structured reinforcement learning. In *Advances in Neural Information Processing Systems*, 2018.
- Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.
- Travis E Oliphant. Python for scientific computing. *Computing in Science & Engineering*, 9(3):10–20, 2007.
- Yann Ollivier. Approximate temporal difference learning is a gradient descent for reversible policies. *arXiv preprint arXiv:1805.00869*, 2018.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 2018.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Ronald Parr, Lihong Li, Gavin Taylor, Christopher Painter-Wakefield, and Michael L Littman. An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2008.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

Jason Papis and Ronald Parr. Pac optimal exploration in continuous space markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2013.

Marek Petrik. An analysis of laplacian methods for value function approximation in mdps. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2007.

David Pfau, Stig Petersen, Ashish Agarwal, David GT Barrett, and Kimberly L Stachenfeld. Spectral inference networks: Unifying deep and spectral learning. In *Proceedings of the International Conference on Learning Representations*, 2019.

Bilal Piot, Matthieu Geist, and Olivier Pietquin. Difference of convex functions programming for reinforcement learning. In *Advances in Neural Information Processing Systems*, 2014.

- Dean A Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural computation*, 3(1):88–97, 1991.
- Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.
- Emmanuel Rachelson and Michail G. Lagoudakis. On the locality of action domination in sequential decision making. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics*, 2010.
- Bohdana Ratitch and Doina Precup. Sparse distributed memories for on-line value-based reinforcement learning. In *Proceedings of the European Conference on Machine Learning*, 2004.
- Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-maroon, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *Transactions on Machine Learning Research*, 2022.
- Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2010.
- H.L. Royden. *Real Analysis*. Prentice Hall, Upper Saddle River, New Jersey 07458, 3 edition, 1968. ISBN 0024041513.
- Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.
- Bruno Scherrer. Should one compute the temporal difference fix point or minimize the bellman residual? the unified oblique projection view. In *Proceedings of the International Conference on Machine Learning*, 2010.

- Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *Proceedings of the International Conference on Learning Representations*, 2021.
- David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.
- Sean R Sinclair, Siddhartha Banerjee, and Christina Lee Yu. Adaptive discretization for episodic reinforcement learning in metric spaces. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(3):1–44, 2019.
- Alec Solway, Carlos Diuk, Natalia Córdova, Debbie Yee, Andrew G Barto, Yael Niv, and Matthew M Botvinick. Optimal behavioral hierarchy. *PLoS Computational Biology*, 10(8):e1003779, aug 2014.
- Zhao Song and Wen Sun. Efficient model-free reinforcement learning in metric spaces. *arXiv preprint arXiv:1905.00475*, 2019.
- Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *Proceedings of the International Conference on Machine Learning*, 2003.
- Kimberly L. Stachenfeld, Matthew Botvinick, and Samuel J. Gershman. Design principles of the hippocampal cognitive map. In *Advances in Neural Information Processing Systems*, 2014.
- Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via non-convex factorization. *IEEE Transactions on Information Theory*, 62(11):6535–6579, 2016.

- Wilson A Sutherland. *Introduction to metric and topological spaces*. Oxford University Press, 2009.
- Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.
- Richard S Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems*, 1995.
- Richard S Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems*, 1996.
- Richard S Sutton and Andrew G Barto. *Reinforcement learning: An Introduction*. MIT Press, 1998.
- Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An Introduction*. MIT Press, 2nd edition, 2018.
- Richard S Sutton, Csaba Szepesvári, Alborz Geramifard, and Michael P Bowling. Dyna-style planning with linear function approximation and prioritized sweeping. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2008.
- Richard S Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M Pilarski, Adam White, and Doina Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, 2011.
- Richard S Sutton, A Rupam Mahmood, and Martha White. An emphatic approach to the problem of off-policy temporal-difference learning. *Journal of Machine Learning Research*, 17(1):2603–2631, 2016.

- R.S. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112: 181–211, 1999.
- Adrien Ali Taiga, Rishabh Agarwal, Jesse Farebrother, Aaron Courville, and Marc G Bellemare. Investigating multi-task pretraining and generalization in reinforcement learning. In *Proceedings of the International Conference on Learning Representations*, 2023.
- Cheng Tang. Exponentially convergent stochastic k-pca without variance reduction. In *Advances in Neural Information Processing Systems*, 2019.
- Yunhao Tang, Zhaohan Daniel Guo, Pierre Harvey Richemond, Bernardo Ávila Pires, Yash Chandak, Rémi Munos, Mark Rowland, Mohammad Gheshlaghi Azar, Charline Le Lan, Clare Lyle, et al. Understanding self-predictive learning for reinforcement learning. *arXiv preprint arXiv:2212.03319*, 2023.
- Jonathan Taylor, Doina Precup, and Prakash Panagaden. Bounding performance loss in approximate mdp homomorphisms. In *Advances in Neural Information Processing Systems*, 2009.
- Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(1):1633–1685, 2009.
- Shantanu Thakoor, Mark Rowland, Diana Borsa, Will Dabney, Rémi Munos, and André Barreto. Generalised policy improvement with geometric policy composition. In *Proceedings of the International Conference on Machine Learning*, 2022.
- Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. In *Advances in Neural Information Processing Systems*, 2021.

- Ahmed Touati, Adrien Ali Taiga, and Marc G Bellemare. Zooming for efficient model-free reinforcement learning in metric spaces. *arXiv preprint arXiv:2003.04069*, 2020.
- Ahmed Touati, J er my Rapin, and Yann Ollivier. Does zero-shot reinforcement learning exist? In *Proceedings of the International Conference on Learning Representations*, 2023.
- Joel A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8, 2015.
- John Tsitsiklis and Benjamin Van Roy. Analysis of temporal-difference learning with function approximation. In *Advances in Neural Information Processing Systems*, 1996.
- Guido Van Rossum and Fred L Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.
- Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1995.
- David Venuto, Sherry Yang, Pieter Abbeel, Doina Precup, Igor Mordatch, and Ofir Nachum. Multi-environment pretraining enables transfer to action limited datasets. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.
- Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.
- Nino Vieillard, Olivier Pietquin, and Matthieu Geist. Munchausen reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.
- C dric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

- Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in science & engineering*, 13(2):22–30, 2011.
- Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8: 279–292, 1992.
- Laurenz Wiskott and Terrence J Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002.
- Tian Ye and Simon S Du. Global convergence of gradient descent for asymmetric low-rank matrix factorization. In *Advances in Neural Information Processing Systems*, 2021.
- Huizhen Yu and Dimitri P Bertsekas. Basis function adaptation methods for cost approximation in mdp. In *Proceedings of the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, 2009.
- Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *Proceedings of the International Conference on Learning Representations*, 2020.
- Shangdong Zhang, Hengshuai Yao, and Shimon Whiteson. Breaking the deadly triad with a target network. In *Proceedings of the International Conference on Machine Learning*, 2021.
- Dongbin Zhao and Yuanheng Zhu. Mec—a near-optimal online reinforcement learning algorithm for continuous deterministic systems. *IEEE Transactions on Neural Networks and Learning Systems*, 26(2):346–356, 2014.