Université de Montréal

Large State Spaces and Self-supervision in Reinforcement Learning.

par

Ahmed Touati

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des arts et des sciences
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en informatique

August, 2021

Université de Montréal
Faculté des arts et des sciences

Cette thèse intitulée:

Large State Spaces and Self-supervision in Reinforcement Learning.

présentée par:

Ahmed Touati

a été évaluée par un jury composé des personnes suivantes:

Yoshua Bengio,	président-rapporteur
Pascal Vincent,	directeur de recherche
Ioannis Mitliagkas,	membre du jury
Matthieu Geist,	examinateur externe

Thèse acceptée le: . . . . . .

To my beloved parents Najoua and Mohammed,
to whom I owe an eternal gratitude.

Résumé

L'apprentissage par renforcement (RL) est un paradigme d'apprentissage orienté agent qui s'intéresse à l'apprentissage en interagissant avec un environnement incertain. Combiné à des réseaux de neurones profonds comme approximateur de fonction, l'apprentissage par renforcement profond (Deep RL) nous a permis récemment de nous attaquer à des tâches très complexes et de permettre à des agents artificiels de maîtriser des jeux classiques comme le Go, de jouer à des jeux vidéo à partir de pixels et de résoudre des tâches de contrôle robotique.

Toutefois, un examen plus approfondi de ces remarquables succès empiriques révèle certaines limites fondamentales. Tout d'abord, il a été difficile de combiner les caractéristiques souhaitables des algorithmes RL, telles que l'apprentissage hors politique et en plusieurs étapes, et l'approximation de fonctions, de manière à obtenir des algorithmes stables et efficaces dans de grands espaces d'états. De plus, les algorithmes RL profonds ont tendance à être très inefficaces en raison des stratégies d'exploration-exploitation rudimentaires que ces approches emploient. Enfin, ils nécessitent une énorme quantité de données supervisées et finissent par produire un agent étroit capable de résoudre uniquement la tâche sur laquelle il est entrainé. Dans cette thèse, nous proposons de nouvelles solutions aux problèmes de l'apprentissage hors politique et du dilemme exploration-exploitation dans les grands espaces d'états, ainsi que de l'auto-supervision dans la RL.

En ce qui concerne l'apprentissage hors politique, nous apportons deux contributions. Tout d'abord, pour le problème de l'évaluation des politiques, nous montrons que la combinaison des méthodes populaires d'apprentissage hors politique et à plusieurs étapes avec une paramétrisation linéaire de la fonction de valeur pourrait conduire à une instabilité indésirable, et nous dérivons une variante de ces méthodes dont la convergence est prouvée. Deuxièmement, pour l'optimisation des politiques, nous proposons de stabiliser l'étape d'amélioration des politiques par une régularisation de divergence hors politique qui contraint les distributions stationnaires d'états induites par des politiques consécutives à être proches les unes des autres.

Ensuite, nous étudions l'apprentissage en ligne dans de grands espaces d'états et nous nous concentrons sur deux hypothèses structurelles pour rendre le problème traitable : les environnements lisses et linéaires. Pour les environnements lisses, nous proposons un algorithme en ligne efficace qui apprend activement un partitionnement adaptatif de l'espace commun en zoomant sur les

régions les plus prometteuses et fréquemment visitées. Pour les environnements linéaires, nous étudions un cadre plus réaliste, où l'environnement peut maintenant évoluer dynamiquement et même de façon antagoniste au fil du temps, mais le changement total est toujours limité. Pour traiter ce cadre, nous proposons un algorithme en ligne efficace basé sur l'itération de valeur des moindres carrés pondérés. Il utilise des poids exponentiels pour oublier doucement les données qui sont loin dans le passé, ce qui pousse l'agent à continuer à explorer pour découvrir les changements.

Enfin, au-delà du cadre classique du RL, nous considérons un agent qui interagit avec son environnement sans signal de récompense. Nous proposons d'apprendre une paire de représentations qui mettent en correspondance les paires état-action avec un certain espace latent. Pendant la phase non supervisée, ces représentations sont entraînées en utilisant des interactions sans récompense pour encoder les relations à longue portée entre les états et les actions, via une carte d'occupation prédictive. Au moment du test, lorsqu'une fonction de récompense est révélée, nous montrons que la politique optimale pour cette récompense est directement obtenue à partir de ces représentations, sans aucune planification. Il s'agit d'une étape vers la construction d'agents entièrement contrôlables.

Un thème commun de la thèse est la conception d'algorithmes RL prouvables et généralisables. Dans la première et la deuxième partie, nous traitons de la généralisation dans les grands espaces d'états, soit par approximation de fonctions linéaires, soit par agrégation d'états. Dans la dernière partie, nous nous concentrons sur la généralisation sur les fonctions de récompense et nous proposons un cadre d'apprentissage non-supervisé de représentation qui est capable d'optimiser toutes les fonctions de récompense.

Abstract

Reinforcement Learning (RL) is an agent-oriented learning paradigm concerned with learning by interacting with an uncertain environment. Combined with deep neural networks as function approximators, deep reinforcement learning (Deep RL) allowed recently to tackle highly complex tasks and enable artificial agents to master classic games like Go, play video games from pixels, and solve robotic control tasks. However, a closer look at these remarkable empirical successes reveals some fundamental limitations. First, it has been challenging to combine desirable features of RL algorithms, such as off-policy and multi-step learning with function approximation in a way that leads to both stable and efficient algorithms in large state spaces. Moreover, Deep RL algorithms tend to be very sample inefficient due to the rudimentary exploration-exploitation strategies these approaches employ. Finally, they require an enormous amount of supervised data and end up producing a narrow agent able to solve only the task that it was trained on. In this thesis, we propose novel solutions to the problems of off-policy learning and exploration-exploitation dilemma in large state spaces, as well as self-supervision in RL.

On the topic of off-policy learning, we provide two contributions. First, for the problem of policy evaluation, we show that combining popular off-policy and multi-step learning methods with linear value function parameterization could lead to undesirable instability, and we derive a provably convergent variant of these methods. Second, for policy optimization, we propose to stabilize the policy improvement step through an off-policy divergence regularization that constrains the discounted state-action visitation induced by consecutive policies to be close to one another.

Next, we study online learning in large state spaces and we focus on two structural assumptions to make the problem tractable: smooth and linear environments. For smooth environments, we propose an efficient online algorithm that actively learns an adaptive partitioning of the joint space by zooming in on more promising and frequently visited regions. For linear environments, we study a more realistic setting, where the environment is now allowed to evolve dynamically and even adversarially over time, but the total change is still bounded. To address this setting, we propose an efficient online algorithm based on weighted least squares value iteration. It uses exponential weights to smoothly forget data that are far in the past, which drives the agent to keep exploring to discover changes.

Finally, beyond the classical RL setting, we consider an agent interacting with its environments without a reward signal. We propose to learn a pair of representations that map state-action pairs to some latent space. During the unsupervised phase, these representations are trained using reward-free interactions to encode long-range relationships between states and actions, via a predictive occupancy map. At test time, once a reward function is revealed, we show that the optimal policy for that reward is directly obtained from these representations, with no planning. This is a step towards building fully controllable agents.

A common theme in the thesis is the design of provable RL algorithms that generalize. In the first and the second part, we deal with generalization in large state spaces either by linear function approximation or state aggregation. In the last part, we focus on generalization over reward functions and we propose a task-agnostic representation learning framework that is provably able to solve all reward functions.

Keywords—Mots-clés

reinforcement learning, markov decision process, artificial agent, off-policy learning, function approximation, exploration-exploitation trade-off, self-supervision, generalization

apprentissage par renforcement, processus de décision markovien, agent artificiel, apprentissage hors-politique, approximation de fonction, compromis exploration-exploitation, auto-supervision, généralisation

Contents

List of Tables

List of Figures

List of acronyms and abbreviations

RL	reinforcement learning
e.g.	*exempli gratia* [for instance]
MDP	Markov decision process
TD	temporal difference
GTD	gradient temporal difference
MSPBE	mean squared projected Bellman error
OFU	optimism in face to the uncertainty
DQN	deep Q-network
TRPO	trust region policy optimization
PPO	proximal policy optimization
CPI	conservative Policy Iteration
KL	Kullback-Leibler
FB	forward-backward representations

Acknowledgements

My PhD was both a humanly and scientifically rich journey. I am incredibly fortunate to have done my PhD at Mila, certainly one of the most thriving environments in the field. I feel greatly grateful to be surrounded by an uncountable number of beautiful humans and brilliant minds. The abovementioned acknowledgments are surely brief and incomplete and won't reflect perfectly my deep gratitude.

To my mother and my father, Najoua and Mohammed, to whom this thesis is dedicated, for their unconditional love, prayers and for raising me to value education. To my sister and brother, Yosra and Khalil, for their support and eternal companionship despite the geographical distance between us.

To Pascal Vincent, for believing in me and for giving me the opportunity to spend these last 5 years at Mila. Being advised by him is a privilege. His scientific rigor, honesty and openness helped me take shape as a researcher. Besides being a remarkable academic mentor, he provided me a big emotional support. He always managed to lift my spirit and helped me to go through my tough times. None of this thesis would have existed without him.

To Doina Precup, for being a source of inspiration. Her introductory course of RL is the reason why I decided to do my PhD in this field. I felt honored when she invited me to attend my first conference RLDM 2017 to discover the active research topics in RL. She is a wonderful woman.

To Pierre Luc Bacon, for all his advice and helpful technical discussions, especially at the early stage of my PhD. He also helped me to know better the RL community by introducing me to many prominent researchers during conferences.

To Joelle Pineau, for hosting me as part-time researcher within her Facebook Research team in Montreal, for all the interesting discussions and thoughts she shared with me. Her role as charismatic leader and great researcher makes her a source of inspiration for many of us.

To Marc G. Bellemare, for offering me the opportunity to do an internship at Google Brain in Montreal under his supervision, for his clarity of thoughts and for always challenging my views.

To Yann Ollivier, for offering me an internship at Facebook research in Paris under his supervision and whom I learnt a disproportionate amount from in the short time. He is remarkable researcher and was fully committed to our project. It was fantastic to collaborate with him.

To Michal Valko, for being a good friend, for all his advice and guidance, for our technical discussions and for helping me to overcome my insecurities.

To Alessandro Lazaric, for giving me a two-hour crash course on online learning in RL, for being my mentor during my Facebook internship and for his willingness to answer my technical questions.

To Matteo Pirotta, for the stimulating discussions and valuable feedback.

To all my collaborators at Mila, for the exciting work we have done together, Chin-Wei Huang, Amy Zhang, Joshua Romoff and Harsh Satija,... to name few of them. Special thanks to Harsh for providing me feedback on almost all my papers and for always showing up at my posters during online events so I don't feel alone in the zoom room.

To my big Montreal family, who made my Montreal life more colorful and warmer. To Max, Pier, Ben and Brett, for our beach volley ball sessions, for all the the craziness and laughter we had together. To Léo, who has been my roommate for almost 4 years, for her support and for making work from home enjoyable. To my other roommates Sylvain and Anna, for all the moments we shared together in our home. To Rim, for being like a sister from the first sight. To Gabriel, Rémi, Gauthier, Hugo, Tom, Adrien, Valentin, Salem, David, Anne-Marie, Victor, Angèle, Lucile, Benoît, Tess, Elodie, for all the experiences and memories we created together. Finally, to Clement for always helping me to go through the toughest moments of my PhD with his enthusiasm and creativity.

To my Friends from Tunisia and France, Omar, Achref, Nouha, Frikha, Tarak, Dali, Lamloum, Qays, Kaabar, Amal, the two Nicolas, Florent, Fadi, Simon, Matthieu, Solomane, Maxime, Rémi, Benjamin whom I keep a strong tie with despite the geographical distance, for their constant support. I'm thankful to be surrounded by such wonderful friends.

Notation

The set of real numbers	$\mathbb{R}$
The set $\{1, 2, \dots, N\}$ for an integer $N > 0$	$[N]$
The set of states	$\mathcal{S}$
The set of actions	$\mathcal{A}$
The discount factor	γ
The reward upon taking the action a in state s	$r(s, a)$
Transitioning probability into s' conditioned on (s, a)	$P(ds' \mid s, a)$
Planning horizon	H
The reward upon taking the action a in state s at stage h	$r_h(s, a)$
Transitioning probability into s' conditioned on (s, a) at stage h	$P_h(ds' \mid s, a)$
Policy	π
The value function and action-value function of policy π	V^π, Q^π
The advantage function for policy π	A^π
Optimal value function and action-value function	$V^\star, Q^\star$
One-step transition operator for policy π	P^π
Bellman evaluation operator for policy π	$\mathcal{T}^\pi$
Bellman optimality operator	$\mathcal{T}$
The discounted state visitation for π and initial distribution ρ	d_ρ^π
Successor states measure	$M^\pi(s, a, ds', da')$
Successor states density	$m^\pi(s, a, s', a')$
The Euclidien norm of vector x	$\|x\|$
The supremum norm of vector x	$\|x\|_\infty$
The Euclidien norm of vector x weighted by the matrix V	$\|x\|_V$
The spectral norm of matrix V	$\|V\|$
The Frobenius norm of the matrix V	$\|V\|_F$
The Kullback-Leibler divergence	D_{KL}
The total variation distance	D_{TV}
Indicator function that returns 1 if $a = b$ and 0 otherwise	$\mathbb{1}_{\{a=b\}}$

1 Introduction

«Intelligence is the computational part of the ability to achieve goals in the world» –John McCarthy

Reinforcement Learning (RL) is a computational approach to solving sequential decision-making problems. Through trial and error an agent must learn to act optimally in an unknown environment in order to maximize its expected utility. A standard setting where a RL agent evolves is illustrated in Figure 1.1: the agent observes the environment's state and a reward associated with the last state transition. It chooses an action. The environment makes a transition to a new state and sends it back to the agent, and the process is repeated. The aim of the agent is to learn to control the environment so as to maximize the long-term total reward. What makes RL different from supervised learning is that the agent receives only partial feedback about his executed actions. Moreover, actions may have a long term effects. For example, a RL agent may take many decisions over multiple steps where immediate rewards are (almost) zero and where more relevant events are rather distant in the future. Such delayed signal makes it challenging to assign credit to temporally distant states and actions that have an effect on the rewards. Finally, the agent's actions affect the subsequent data it receives. The agent needs sometimes to sacrifice its current utility and seek actions that reveal new information about its environment. This will prevent it from prematurely exploiting early limited knowledge and falling into local optima.

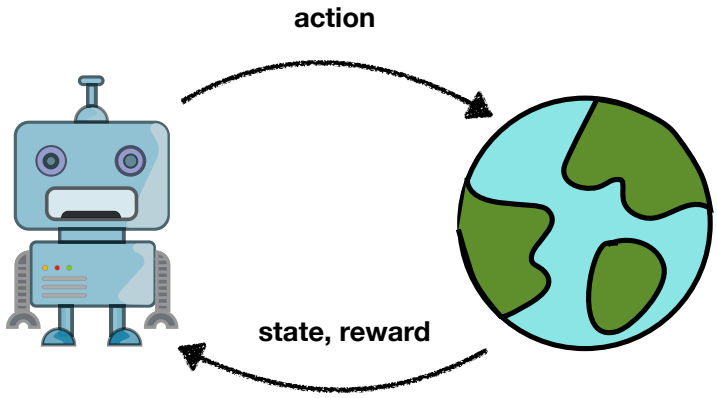

Figure 1.1: Standard Reinforcement Learning setting

While RL remained on the margins of the broader artificial intelligence (AI) community until relatively recently, today, RL flourishes as a vital learning paradigm that captures many important AI applications, including self-driving cars, robots, and adaptive medical treatments. This recent success is partly due to the deployment of flexible function approximation schemes (such as the use of deep neural networks in Deep RL) to deal with large state spaces. However, a closer look at these impressive empirical results reveals some fundamental limitations:

- It has been challenging to combine desirable features of RL algorithms, such as off-policy and multi-step learning with function approximation in a way that leads to both stable and efficient algorithms.
- Deep RL algorithms tend to be very sample inefficient due to the rudimentary exploration-exploitation strategies these approaches employ.
- Furthermore, They require an enormous amount of supervised data and end up producing a narrow agent able to solve only the task that it is trained on.

Addressing the aforementioned challenges and bridging the gap between practise and theory in RL are crucial steps to bring RL to real life and fulfill its promise, and this is what inspired my research.

This thesis is structured as follows: The next section provides a synthetic overview of the research contributions that will be detailed in subsequent chapters. Readers less familiar with RL and its terminology and open challenges will benefit from first reading the background presented in Chapter 2, which attempts to be a compact formal introduction to RL and to the notions this thesis builds on. Each of the following chapters 3 to 7 corresponds to a published or submitted research paper. Chapter 8 concludes the thesis, discussing future research directions.

As this is a thesis by article, the review and discussion of the literature most relevant to each contribution is to be found, in its context, within that contribution's chapter, as in the corresponding article, rather than in a separate dedicated literature review chapter.

1.1 Research Contributions

1.1.1 Off-Policy Learning

Off-policy learning is concerned with evaluating or optimizing a target policy based on data from a different behavior policy. This can provide many benefits: efficient parallel exploration and, reuse of past experience with experience replay and, in many practical contexts, learning form data produced by policies that are currently deployed, but which we want to improve.

However, we show in *«Convergent Tree Backup and Retrace with Function Approximation»* [Touati et al., 2018] (Chapter 3) that combining popular off-policy and multi-step learning methods, for **policy evaluation**, with linear value function parameterization could lead to undesirable instability. Then, we derive a provably convergent variant of these methods by converting the problem into a primal-dual saddle point problem. Finally, we provide their finite sample analysis.

In another work *« Stable Policy Optimization via Off-Policy Divergence Regularization»* Touati et al. [2020c] (Chapter 4), we ask a different question:

Given a stream of experience generated by a given policy, how could we improve it in a safe and stable way?

For that, we revisit the foundations of **conservative approaches to policy optimization** that optimize a surrogate objective that can provide local improvements to the current policy at each iteration. We highlight that these methods rely solely on constraining immediate action probabilities and argue that the latter might not be enough and that we should rather reason about the long-term effect of the policies on the distribution of the future states. In particular, we directly consider the divergence between **state-action occupancy measures** induced by successive policies and use it as a regularization term added to the surrogate objective. This regularization term is itself optimized in an adversarial and off-policy manner. Finally, we empirically show that our proposed method can have a beneficial effect on stability and improve final performance in benchmark high-dimensional control tasks.

1.1.2 Exploration v.s. Exploitation Dilemma

An RL agent, through trial and error, must learn to act optimally in an unknown environment to maximize its expected utility. Exploration refers to the execution of diversified actions to collect data that provide a well-rounded characterization of the environment. Thus, an efficient learning requires balancing exploration (to gain more knowledge) and exploitation (acting optimally according to the available knowledge).

While a wealth of research has been developed to design systematic exploration strategies in small state-space problems (known as tabular problems), their theoretical measure guarantee typically scales with the number of discrete states and the number of discrete actions. This precludes applying them to arbitrarily large state-action spaces.

An appealing challenge is to combine exploration strategies with generalization methods in a way that leads to both provable sample efficient and computationally efficient RL algorithms for large-scale problems. This would require some **structural assumptions**.

A simple way to ensure generalization over states is to aggregate them into a finite set of meta-states and run a tabular exploration mechanism on the latter. In this direction, we proposed in «*Zooming for Efficient Model-free Reinforcement Learning in Metric Spaces*» [Touati et al., 2020b] (Chapter 5) to actively explore the state-action space by learning on the-fly an **adaptive partitioning** that takes into account the shape of the optimal value function. When the state-action space is assumed to be a **compact metric space**, such adaptive discretization based algorithms yield sublinear regret.

Another structural assumption, that received attention in the recent literature, is when both reward and transition dynamics are linear functions with respect to a given feature mapping. This assumption enables the design of efficient algorithms with a linear representation of the action value function. In our work «*Efficient learning in non-stationary linear Markov decision processes*» [Touati and Vincent, 2020a] (Chapter 6), we add **exogenous non-stationarity** where the environment is now allowed to evolve dynamically and even adversarially over time, but the total change is still bounded. This is a more challenging setting, since what has been learned in the past may be obsolete in the present. To address this setting, our proposed algorithm is based on weighted least squares value iteration that uses exponential weights to smoothly forget data that are far in the past, which drives the agent to keep exploring to discover changes. We prove that this algorithm achieves state-of-the-art online performance.

1.1.3 Unsupervised Learning in RL

In standard RL, the agent is given a training signal from the environment in the form of a scalar reward and aims to maximize the expected cumulative rewards along its trajectories. This strategy ends up producing a narrow agent that is only able to solve the single task that it is trained on. Furthermore, it is arguably not how humans learn in real life. For example, a child might get supervisions from his parents in various forms such as verbal instructions or applause. But most of the time, he is exploring the world through curiosity and play and gaining knowledge about his surroundings.

In our work «*Learning One Representation to Optimize All Rewards*» [Touati and

Ollivier, 2021] (Chapter 7), we ask this question:

Given an environment without reward information, is it possible to learn and store a compact object that, for any reward function specified later, provides the optimal policy for that reward, with a minimal amount of additional computation?

To address the question above, we propose to learn a pair of **representations** F (for forward) and B (for backward) that map state-action pairs to some latent space. During the unsupervised learning phase, these representations are trained using **reward-free interactions** such that $F^\top B$ encodes long-range relationships between states and actions, via a **predictive occupancy map**. At test time, once a reward function is revealed, we show that the optimal policy for that reward is directly obtained from these representations, with no planning. Empirically, our approach compares well to goal-oriented RL algorithms on discrete and continuous mazes, pixel-based MsPacman (an Atari game), and a virtual robot arm. We also illustrate how the agent can immediately adapt to new tasks beyond goal-oriented RL.

1.2 List of Excluded Contributions

During my PhD, I produced other contributions on various machine learning topics, that are not included in this thesis.

- **Reinforcement Learning:**
  - Joshua Romoff, Peter Henderson, David Kanaa, Emmanuel Bengio, **Ahmed Touati**, Pierre-Luc Bacon, Joelle Pineau. TDprop: Does Jacobi Preconditioning Help Temporal Difference Learning?. AAMAS 2021. [Romoff et al., 2021]
  - **Ahmed Touati**, Pascal Vincent. Sharp Analysis of Smoothed Bellman Error Embedding. ICML 2020 Workshop on Theoretical Foundations of Reinforcement Learning. [Touati and Vincent, 2020b]
  - Zilun Peng*, **Ahmed Touati***, Pascal Vincent and Doina Precup. SVRG for Policy Evaluation with Fewer Gradient Evaluations. IJCAI 2020. [Peng et al., 2019]
  - Joshua Romoff, Peter Henderson, **Ahmed Touati**, Yann Ollivier, Emma Brunskill, Joelle Pineau. Separating value functions across time-scales. ICML 2019. [Romoff et al., 2019]

- Nan Rosemary Ke, Amanpreet Singh, **Ahmed Touati**, Anirudh Goyal, Yoshua Bengio, Devi Parikh and Dhruv Batra. Learning Dynamics Model in Reinforcement Learning by Incorporating the Long Term Future. ICLR 2019. [Ke et al., 2019]
- **Ahmed Touati**, Harsh Satija, Joshua Romoff, Joelle Pineau, Pascal Vincent. Randomized Value Functions via Multiplicative Normalizing Flows. UAI 2019. [Touati et al., 2020a]

- **Bayesian Deep Learning:**
  - Chin-Wei Huang, **Ahmed Touati**, Pascal Vincent, Gintare Karolina Dziugaite, Alexandre Lacoste, Aaron Courville. Stochastic Neural Network with Kronecker Flow. AISTATS 2020. [Huang et al., 2020]
  - Chin-Wei Huang, **Ahmed Touati**, Laurent Dinh, Michal Drozdzal, Mohammad Havaei, Laurent Charlin, Aaron Courville. Learnable Explicit Density for Continuous Latent Space and Variational Inference. ICML 2017 workshop. [Huang et al., 2017a]

- **Generative Modeling:**
  - Gabriel Huang, Hugo Berard, **Ahmed Touati**, Gauthier Gidel, Simon Julien-Lacoste. Adversarial Divergences are Good Task Losses for Generative Modeling. Under review at Journal of Machine Learning Research. [Huang et al., 2017b]

- **Convex Optimization:**
  - Rémi Le Priol, **Ahmed Touati** and Simon Lacoste-Julien. Adaptive Stochastic Dual Coordinate Ascent for Conditional Random Fields. NIPS 2017 Workshop on Optimization for Machine Learning.

2 Background

2.1 Discounted Markov Decision Process

2.1.1 The Model

Markov Decision Processes [Puterman, 1994] are a mathematical tool for modeling sequential decision-making problems where a decision maker or an agent interacts with a system in a sequential fashion. A discounted Markov Decision Process (MDP) is a tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, transition probabilities P mapping state-action pairs to distributions over next states, immediate reward function r, and a discount fator $\gamma \in [0, 1)$. The reward could be stochastic. If $\mathcal{S}$ is discrete, for each (s, a), $P(s' \mid s, a)$ is a probability mass function on $s' \in \mathcal{S}$. For general $\mathcal{S}$, $P(ds' \mid s, a)$ is a probability measure on $s' \in \mathcal{S}$.

For the ease of exposition in this chapter, we restrict our attention to finite MDPs ($\mathcal{S}$ and $\mathcal{A}$ are finite). We also assume that the reward function is bounded, $r : \mathcal{S} \times \mathcal{A} \to [0, r_{\max}]$ for some positive scalar $r_{\max}$.

An agent can select its actions at any stage based on the observed history. A rule describing the way the actions are selected is called a policy. Starting from an initial state s_0 and executing the policy, the agent generates a random sequence of state-action-reward $\{s_0, a_0, r_0, s_{k+1}, \ldots s_t, a_t, r_t, s_{t+1} \ldots\}$ where $r_t = r(s_t, a_t)$ and $s_{t+1} \sim p(\cdot \mid s_t, a_t)$. The return underlying a behavior is defined as the total discounted sum of the rewards incurred:

$$\mathcal{R} = \sum_{t=0}^{\infty} \gamma^t r_t \tag{2.1.1}$$

The goal of the agent is to choose a behavior that maximizes the expected return. Such a maximizing policy is said to be optimal. It is known that an optimal policy can be *memoryless* in the sense that it does not need the whole observed history and may only depend on the current state. Moreover, it is known that to be optimal over an infinite horizon setting, a policy can be *stationary* in the sense that it does not depend on the time that a state was encountered.

2.1.2 Policies and Value Functions

As stated above, we will restrict our attention to memoryless and stationary policies. Formally, we define a policy as a mapping from states to probabilities of selecting each possible action. We denote by $\pi(a \mid s)$ the probability of choosing action a in state s under the policy $\pi : \mathcal{S} \to \text{Prob}(\mathcal{A})$, where $\text{Prob}(\mathcal{A})$ denotes the space of probability distribution over actions.

An action-value function, know also as Q-function, is a function of state-action pairs that estimate how good it is for the agent to be in a given state-action pair. Thus, it is a way to evaluate policy and compare different policies. The value of a state-action pair (s, a) under a policy π, denoted $Q^\pi(s)$, is the expected return when starting in state s, taking action a and following π thereafter:

$$Q^\pi(s,a) \triangleq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \;\middle|\; (s_0, a_0) = (s,a), \pi\right] \tag{2.1.2}$$

Where the expectation is over trajectories drawn by executing π: $\{(s_t, a_t, r_t, s_{t+1})\}_t$ where $a_t \sim \pi(.|s_t), \quad r_t = r(s_t, a_t), \quad \text{and} \quad s_{t+1} \sim p(\cdot \mid s_t, a_t)$. Similarly, the value of a state s under a policy π, denoted $V^\pi(s)$, is the expected return when starting in state s and following π thereafter:

$$V^\pi(s) \triangleq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \;\middle|\; s_0 = s, \pi\right] \tag{2.1.3}$$

Value function and action-value function are related as follows:

$$V^\pi(s) = \mathbb{E}_{a \in \pi(\cdot|s)}[Q^\pi(s,a)] = \sum_{a \in \mathcal{A}} \pi(a \mid s) Q^\pi(s,a) \tag{2.1.4}$$

A fundamental property of action-value functions used throughout reinforcement learning is that they satisfy the following recursive relationship:

$$\begin{aligned}
Q^\pi(s,a) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \;\middle|\; (s_0, a_0) = (s,a), \pi\right] \\
&= r(s,a) + \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{t+1} r_{t+1} \;\middle|\; (s_0, a_0) = (s,a), \pi\right] \\
&= r(s,a) + \gamma \sum_{s'} P(s' \mid s,a) \underbrace{\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \;\middle|\; s_0 = s', \pi\right]}_{V^\pi(s')} \\
&= r(s,a) + \gamma \sum_{s',a'} P(s' \mid s,a) \pi(a' \mid s') Q^\pi(s',a')
\end{aligned} \tag{2.1.5}$$

This is the *Bellman equation* for Q^π. It expresses a relationship between the value of a state-action pair and the values of its successor states and actions. Similarly, the Bellman equation for V^π is defined as follows:

$$V^\pi(s) = \sum_a \pi(a \mid s) r(s,a) + \gamma \sum_{s',a} \pi(a \mid s) P(s'|s,a) V^\pi(s') \tag{2.1.6}$$

2.1.3 Optimal Policies and Optimal Value Functions

The optimal value functions, denoted by $V^\star$ and $Q^\star$ respectively, are defined by:

$$V^\star(s) \triangleq \max_\pi V^\pi(s), \tag{2.1.7}$$

$$Q^\star(s,a) \triangleq \max_\pi Q^\pi(s,a) \tag{2.1.8}$$

A remarkable property of MDPs is that there exists a stationary and deterministic policy π that simultaneously maximizes $V^\pi(s)$ for all $s \in \mathcal{S}$ [Puterman, 1994]. We refer to such a π as an optimal policy. The optimal value and action-value functions are connected by the following equations:

$$V^\star(s) = \max_{a \in \mathcal{A}} Q^\star(s,a) \tag{2.1.9}$$

$$Q^\star(s,a) = r(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V^\star(s') \tag{2.1.10}$$

The latter equation is known as the *optimal Bellman equation*. We know that for all policies π, $V^\pi(s) = \sum_a \pi(a \mid s) Q^\pi(s,a) \quad \forall s \in \mathcal{S}$. In order latter equation holds for optimal value function $V^\star$, $\pi^\star(a \mid s) > 0$ only if $Q^\star(s,a) = \max_{a' \in \mathcal{A}} Q^\star(s,a')$, which simply corresponds to choosing the action of maximum value in each state. Therefore, the *greedy* policy with respect to $Q^\star$ is an optimal policy.

2.2 Episodic Markov Decision Processes

We presented in the last section the infinite-horizon discounted setting of MDPs where the return of a trajectory is given by $\sum_{t=0}^{\infty} \gamma^t r_t$. Popular alternative choices include the finite-horizon undiscounted setting (return of a trajectory is $\sum_{t=1}^{T} r_t$ with some finite horizon $H < \infty$) and the infinite-horizon average reward setting (return is $\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} r_t$). The latter case often requires additional conditions on the transition dynamics (such as ergodicity) so that values can be well-defined.

A finite horizon MDP is a tuple $(\mathcal{S}, \mathcal{A}, P, r, H)$ where $\mathcal{S}$ and $\mathcal{A}$ are the state and action space, H is the planning horizon i.e number of steps in each episode,

$P = \{P_h\}$ is a collection of the transition probabilities such that $P_h(\cdot|s,a)$ gives the distribution over next states if action a is taken at state s at step $h \in [H]$, $r = \{r_h\}$ is a collection of reward functions such that $r_h(s,a)$ is the reward of taking action a at state s at time step h.

For any step $h \in [H]$ and $(s,a) \in \mathcal{S} \times \mathcal{A}$, the action-value function of a non-stationary policy $\pi = (\pi_1, \ldots, \pi_H)$ is defined as $Q_h^\pi(s,a) = r_h(s,a) + \mathbb{E}\left[\sum_{i=h+1}^{H} r_i(s_i,a_i)) \,\middle|\, (s_h,a_h) = (s,a), \pi\right]$, and the value function is $V_h^\pi(s) = \sum_a \pi_h(a \mid s) Q_h^\pi(s,a)$. As the horizon is finite, there always exists an optimal policy $\pi^\star$ whose value and action-value functions are defined as $V_h^\star(x) = V_h^{\pi^\star}(s) = \max_\pi V_h^\pi(s)$ and $Q_h^\star(s,a) = Q_h^{\pi^\star}(s,a) = \max_\pi Q_h^\pi(s,a)$. Both Q^π and $Q^\star$ can be conveniently written as the result of the following Bellman equations

$$Q_h^\pi(s,a) = r_h(s,a) + \sum_{s'} P_h(s' \mid s,a) V_{h+1}^\pi(s'), \tag{2.2.1}$$

$$Q_h^\star(s,a) = r_h(s,a) + \sum_{s'} P_h(s' \mid s,a) V_{h+1}^\star(s'), \tag{2.2.2}$$

where $V_{H+1}^\pi(s) = V_{H+1}^\star(s) = 0$ and $V_h^\star(s) = \max_{a \in \mathcal{A}} Q_h^\star(s,a)$, for all $s \in \mathcal{S}$.

2.3 Dynamic Programming

«An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision» –Bellman's Principle of Optimality [Bellman, 1957]

The term dynamic programming (DP) refers to a collection of algorithms that can be used to compute optimal policies given a perfect model of the environment as a Markov decision process (MDP). DP leverages the structure of MDP to simplify the problem solving. While DP has limited practical importance as it assumes access to a perfect model, it remains of great theoretical interest and gives valuable insights for the design of RL algorithms.

2.3.1 Bellman Operators

Many DP algorithms operate in the value function space by repeatedly applying transformations. Therefore, it is convenient to think of the latter as operators. If we consider action-value function as mapping : $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$. An operator $\mathcal{B}$ over action-value functions then takes a function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and maps it to a function over the same domain, $\mathcal{B}Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.

Let us first define the transition operator $\mathcal{P}^\pi$ for a policy π:

$$\mathcal{P}^\pi Q(s,a) \triangleq \sum_{s',a'} P(s' \mid s,a)\pi(a' \mid s')Q(s',a'). \tag{2.3.1}$$

The spectral radius of $\mathcal{P}^\pi$, as stochastic matrix, is equal to one. This implies that the *Neuman expansion* $\sum_{t=0}^{\infty} \gamma^t (\mathcal{P}^\pi)^t$ is convergent and is equal to $(I - \gamma\mathcal{P}^\pi)^{-1}$. Therefore, we can write Q^π as

$$Q^\pi = \sum_{t=0}^{\infty} \gamma^t (\mathcal{P}^\pi)^t r = (I - \gamma P^\pi)^{-1} r \tag{2.3.2}$$

Now, we can write The *Bellman evaluation operator* over Q-functions, which corresponds to applying (2.1.5) to all state-action pairs, as for any $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$,

$$\mathcal{T}^\pi Q \triangleq r + \gamma \mathcal{P}^\pi Q \tag{2.3.3}$$

Similarly to this evaluation operator, we can define the *Bellman optimality operator* as, for any $Q : \mathcal{S} \times \mathcal{A} \times \mathbb{R}$,

$$\mathcal{T} Q \triangleq r + \max_\pi \mathcal{P}^\pi Q \tag{2.3.4}$$

Both operators share nice properties that provide the basis for DP algorithms such as *Value Iteration* and *Policy Iteration*. First, the Q-function of Q^π of the policy π is a *fixed point* of $\mathcal{P}^\pi$ while the optimal Q-function $Q^\star$ is a fixed point of $\mathcal{T}$ i.e $\mathcal{T}^\pi Q^\pi = Q^\pi$ and $\mathcal{T} Q^\star = Q^\star$. Moreover, Both operators are γ-*contraction* in supremum norm l_∞ as for any pair of functions $Q_1, Q_2 : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, we have,

$$\|\mathcal{T}^\pi Q_1 - \mathcal{T}^\pi Q_2\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty, \tag{2.3.5}$$

$$\|\mathcal{T} Q_1 - \mathcal{T} Q_2\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty. \tag{2.3.6}$$

Finally, both operators are *monotone* in the sense that for any pair of functions Q_1 and Q_2 such that $Q_1 \leq Q_2$ component-wise, we have $\mathcal{T}^\pi Q_1 \leq \mathcal{T}^\pi Q_2$ and $\mathcal{T} Q_1 \leq \mathcal{T} Q_2$.

2.3.2 Value Iteration

The *Value Iteration* (VI) [Bellman, 1957] algorithm operates on the Q-function space. Starting with an arbitrary, initial Q-function estimate, it repeatedly applies the Bellman optimality operator $\mathcal{T}$ so that the Q-function estimate approaches the optimal Q-function in the limit. A basic form of VI is given in algorithm 1. The behaviour of VI can be analyzed by the Banach fixed point

Algorithm 1 Value Iteration Algorithm

Input: MDP $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$
1: Initialize a value Q_0
2: **for** k = 1 ... K **do**
3: $\quad Q_{k+1} = \mathcal{T} Q_k$
4: **end for**
5: Return the greedy policy $\pi_K(s) = \arg\max_a Q_K(s, a)$

theorem that states, if $\mathcal{X}$ is a complete metric space[1] with a contraction mapping $\mathcal{B} : \mathcal{X} \to \mathcal{X}$, then $\mathcal{B}$ admits a unique fixed point and the sequence $x, \mathcal{B}x, \mathcal{B}^2x, \ldots$ converge to that fixed point for any $x \in \mathcal{X}$. In particular, it shows that the optimal Q-function $Q^\star$ is indeed the unique fixed point of $\mathcal{T}$ and repeated applications of $\mathcal{T}$ are guaranteed to yield it. Furthermore, it can be shown that VI converges at a geometric rate. In fact, we have for any initial function Q,

$$\begin{aligned} \|\mathcal{T}^k Q - Q^\star\|_\infty &= \|\mathcal{T}(\mathcal{T}^{k-1} Q) - \mathcal{T} Q^\star\|_\infty \\ &\leq \gamma \|\mathcal{T}^{k-1} Q - Q^\star\|_\infty \\ &\leq \ldots \leq \gamma^k \|Q - Q^\star\|_\infty \end{aligned}$$

2.3.3 Policy Iteration

Another classic DP approach to solve MDPs is *Policy Iteration* (PI). This algorithm 2 directly searches the optimal policy by interleaving between a policy evaluation step and a policy improvement step. Starting with an arbitrary policy, the algorithm proceeds in an iterative way. At iteration $k > 0$, it computes the action-value function Q^{π_k} of π_k. Next, it improves the policy by updating the policy to be the greedy policy with respect to Q^{π_k} i.e $\pi_{k+1}(s) = \arg\max_a Q^{\pi_k}(s, a)$ for all $s \in \mathcal{S}$. In terms of the Bellman operators, this can be equivalently expressed as $\mathcal{T}^{\pi_{k+1}} Q^{\pi_k} = \mathcal{T} Q^{\pi_k}$. We have that

$$Q^{\pi_k} = \mathcal{T}^{\pi_k} Q^{\pi_k} \leq \mathcal{T} Q^{\pi_k} = \mathcal{T}^{\pi_{k+1}} Q^{\pi_k} \text{ (component-wise).} \tag{2.3.7}$$

By applying the operator $\mathcal{T}^{\pi_{k+1}}$ on both sides and using the monotonicity property, we obtain for any integer $n > 0$

$$Q^{\pi_k} \leq \mathcal{T}^{\pi_{k+1}} Q^{\pi_k} \leq \ldots \leq (\mathcal{T}^{\pi_{k+1}})^n Q^{\pi_k} \text{ (component-wise).} \tag{2.3.8}$$

[1] Intuitively, for a metric space to be complete means that there are no "points missing" from it. Bounded real-valued functions can be shown to be complete, and therefore the space of Q-functions with Q equipped with the l_∞-norm is a complete metric space.

Algorithm 2 Policy Iteration Algorithm

Input: MDP $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$
1: Initialize a policy π_1
2: **for** k = 1 ... K **do**
3: given π_k, compute Q^{π_k} /* Policy Evaluation
4: $\pi_{k+1}(s) = \arg\max_a Q^{\pi_k}(s,a)$ /* Policy Improvement
5: **end for**
6: Return the last policy π_K

By applying the Banach fixed-point theorem on the contraction $\mathcal{T}^{\pi_{k+1}}$, we obtain $\lim_{n\to}(\mathcal{T}^{\pi_{k+1}})^n Q^{\pi_k} = Q^{\pi_{k+1}}$. Which implies that $Q^{\pi_k} \leq Q^{\pi_{k+1}}$. Therefore, PI generates a sequences of policies with non-decreasing performance. Since we have now $\mathcal{T}Q^{\pi_k} = \mathcal{T}^{\pi_{k+1}}Q^{\pi_k} \leq Q^{\pi_{k+1}} \leq Q^\star$, we obtain

$$\begin{aligned}\|Q^{\pi_{k+1}} - Q^\star\|_\infty &\leq \|\mathcal{T}^{\pi_{k+1}}Q^{\pi_k} - Q^\star\|_\infty \\ &= \|\mathcal{T}Q^{\pi_k} - \mathcal{T}Q^\star\|_\infty \\ &\leq \gamma\|Q^{\pi_k} - Q^\star\|_\infty \\ &\leq \ldots \leq \gamma^{k+1}\|Q^{\pi_1} - Q^\star\|_\infty\end{aligned}$$

Thus, PI converge at a linear rate.

2.4 Temporal Difference Learning

«TD learning is learning a prediction from another, later, learned prediction, i.e, learning a guess from a guess» –Richard Sutton at Montreal summer school, 2017.

Dynamic programming algorithms assume that both the transition probabilities P and the reward function r are explicitly known. On the other hand, in the general reinforcement learning setting, this information is not available to the agent which needs a direct interaction with the environment in order to solve it.

For value function estimation at a given state, a straightforward alternative is to approximate the expectation of the random return by an average over independent trajectories started from the given state. This is an instance to the so-called *Monte-Carlo* method. However, this technique faces several issues: the estimator tends to have high variance, it is computed only when all trajectories are terminated, and it assumes that we are able to reset the environment to some particular state.

To address these shortcomings, *temporal difference* (TD) learning [Sutton, 1988] has been proposed and has become the main building block of many RL algorithms. TD learning approaches combine ideas from both Monte Carlo and DP. Similarly to Monte Carlo, they learn from raw experiences without access to an explicit model of the environment. Similarly to DP, they update estimates of the values of states based on estimates of the values of successor states. This latter is referred as *bootsrapping*.

2.4.1 Policy Evaluation

For the problem of policy evaluation, TD learning, in its simplest form TD(0), updates the value of a state-action pair (s_t, a_t) towards an estimate of the return $r_t + \gamma Q(s_{t+1}, a_{t+1})$, called the TD target, where r_t is the reward given by the environment at time t, s_{t+1} is the state reached after the transition, and a_{t+1} is the action selected by the policy being evaluated. More precisely, TD(0) update rule is as follows :

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t(s_t, a_t)\big(r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\big), \tag{2.4.1}$$

where $\alpha_t(s, a) \in (0, 1)$ is the learning rate associated to the state-action pair (s, a) at time t. The term $r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$ is called the TD error, and is usually denoted by δ_t. We have in expectation:

$$\mathbb{E}\left[\delta_t \,\middle|\, (s_t, a_t) = (s, a)\right] = (\mathcal{T}^\pi Q - Q)(s, a), \forall (s, a) \tag{2.4.2}$$

Therefore, TD(0) can be interpreted as a stochastic approximation scheme for solving a fixed point of the Bellman equation. Contraction property together with general results from stochastic approximation theory can then be used to show asymptotic convergence under the minimal assumption that for every state-action pair (s, a), the sequence of step-size $\{\alpha_t(s, a)\}_t$ satisfies Robbins-Monro conditions i.e $\sum_t \alpha_t(s, a) = \infty$ and $\sum_t \alpha_t(s, a)^2 < \infty$ [Jaakkola et al., 1994]. Since the learning rate α_t is less than 1, the first condition $\sum_t \alpha_t(s, a) = \infty$ implies that every state and every action should be visited infinitely often.

While the Monte Carlo method has no bias but could suffer from large variance, TD(0), a one-step TD, has a small variance but may introduce bias. We can extend both methods by n-step TD which can shift from one to the other smoothly as needed.

The target for an arbitrary n-step update is the n-step return:

$$R_{t:t+n} = r_t + \gamma r_{t+1} + \cdots + \gamma^{n-1} r_{t+n-1} + \gamma^n Q(s_{t+n}, a_{t+n}). \tag{2.4.3}$$

All n-step returns can be considered approximations to the full return, truncated after n steps and then corrected for the remaining missing terms by $Q(s_{t+n}, a_{t+n})$.

Therefore, n-step methods enable bootstrapping to occur over multiple steps. Now we note that a valid update can be done not only toward any n-step return, but also toward any average of n-step returns. A particular and popular way of averaging n-step updates is to weight each n-step updates by λ^{n-1}, where $\lambda \in [0, 1]$ and then normalize it by a factor of $1 - \lambda$ to ensure that the weights sum to 1. This particular average is called the λ-return and is defined as

$$R_t^\lambda \triangleq (1-\lambda) \sum_{n=0}^{\infty} \lambda^{n-1} R_{t:t+n} \tag{2.4.4}$$

TD(λ) algorithm updates the value function estimates towards this λ-return as follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t(s_t, a_t) \left(R_t^\lambda - Q(s_t, a_t) \right) \tag{2.4.5}$$

$\lambda = 0$ corresponds to the TD(0) (one-step TD) update, and $\lambda \to 1$ removes the recursion on Q-function estimate, and recover the Monte Carlo estimate. λ controls clearly the amount of sampling versus bootstrapping. Therefore, it is a trade-off parameter between bias and variance. We have that $\mathbb{E}\left[R_t^\lambda\right] = \mathcal{T}_\lambda^\pi V$ where $\mathcal{T}_\lambda^\pi$ is the Bellman λ-operator defined as

$$\mathcal{T}_\lambda^\pi \triangleq (1-\lambda) \sum_{n=0}^{\infty} \lambda^n (\mathcal{T}^\pi)^{n+1} \tag{2.4.6}$$

Q^π remains the fixed point of $\mathcal{T}_\lambda^\pi$. Therefore, TD(λ) could be interpreted as a stochastic approximation scheme for finding the fixed point of the operator $\mathcal{T}_\lambda^\pi$.

2.4.2 Policy Learning

So far, we dealt with TD learning for the problem of approximating the value function of a fixed policy. Now, we present the Q-learning algorithm [Watkins and Dayan, 1992], one natural extension of TD learning to control problems, where the goal is to learn an effective policy from data. Given a transition (s_t, a_t, r_t, s_{t+1}) at time t, Q-learning's update rule is as follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t(s_t, a_t) \left(r_t + \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right). \tag{2.4.7}$$

If we consider the TD error $\delta_t = r_t + \max_a Q(s_{t+1}, a) - Q(s_t, a_t)$, we have in expectation:

$$\mathbb{E}\left[\delta_t \mid s_t = s, a_t = a\right] = (\mathcal{T}Q - Q)(s, a), \forall (s, a) \tag{2.4.8}$$

Q-learning is then a stochastic approximation scheme for finding the fixed point of the optimality Bellman operator $\mathcal{T}$. The sequence of estimates is showed to converge to $Q^\star$ with probability one under the assumption that for every

state-action pair (s, a) the sequence of step-sizes $\{\alpha_t(s, a)\}_t$ satisfies Robbins-Monro conditions [Jaakkola et al., 1994]. This condition requires that every state-action pair is visited infinitely.

While TD(0), presented earlier, requires that actions are executed by the fixed policy that we would like to evaluate, Q-learning allows us to learn from any stream of experiences. Q-learning is thus an *off-policy* method. The commonly used strategies in practice is to sample the actions following the ϵ-greedy or the Botzmann policy with respect to the current estimate of the Q-value. While such sampling strategies ensure the full coverage of the state-action space in the limit of infinite time and eventually lead to asymptotic convergence, a more directed exploration strategy would be critical to achieve a good performance in a reasonable amount of time.

2.5 Function Approximation

When the state space is large, which is the case of most domains of interest, it is not feasible to keep a separate value for each state in the memory. In order to obtain generalization between different state-action pairs, value functions or policies should be represented in a compact functional form.

2.5.1 Value-based Methods

Value-based method learn parametrized value-functions. A traditional choice is the linear function approximation of the form:

$$Q(s, a) = \theta^\top \phi(s, a) \ ,$$

where $\theta \in \Theta \subset \mathbb{R}^d$ is a weight vector and $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ is a feature map from a state-action pairs to a given d-dimensional feature space.

The natural extension of Q-learning to function approximation with parametric form $\{Q_\theta, \theta \in \mathbb{R}^d\}$ is:

$$\theta_{t+1} = \theta_t + \alpha_t \delta_t \nabla_\theta Q_{\theta_t}(s_t, a_t) \tag{2.5.1}$$

where $\delta_t = r_r + \gamma \max_a Q_{\theta_t}(s_{t+1}, a) - Q_{\theta_t}(s_t, a_t)$. Note that with linear function, $\nabla_\theta Q_{\theta_t}(s_t, a_t) = \phi(s_t, a_t)$ and when features are defined by one-hot encoding vectors, we recover the Q-learning algorithm in the tabular case. Moreover, if we ignore the effect of θ on the target $y_t = r_r + \gamma \max_a Q_{\theta_t}(s_{t+1}, a)$, the update rule looks like a stochastic gradient descent on:

$$\mathbb{E}\left[(Q_\theta(s, a) - y)^2\right] \tag{2.5.2}$$

However, we include only a part of the gradient by taking into account the effect of changing the weight vector θ_t on the estimate Q_{θ_t}, but ignore its effect, via the bootstrapping, on the target y_t. Such methods are called *semi-gradient* methods.

Although update rule 2.5.1 is commonly used in practice, not much can be said regarding its convergence properties. In fact, we know that Q-learning with linear function approximation may diverge. Some properties of combining temporal difference with function approximation will be extensively studied in our first contribution.

The limitation of linear functions is that we need to handcraft feature extractors for each domain. This could be painful and time-consuming especially in high-dimensional domains or when a good prior for feature design is not available.

This partially motivates the recent trend of combining reinforcement learning with deep leaning, which is known as deep reinforcement learning (deep RL). Deep RL opens up new applications as it allows us solve a wide range of complex decision-making tasks that were previously out of reach for a machine.

Deep neural networks are made up of multiple processing layers. Each layer consists in an affine transformation followed by non-linear activations. The sequence of non-linear transformations allows to learn different levels of abstractions. This enables representations learning and obviates the need to handcraft features to describe the inputs.

Deep Q-Networks (DQN) [Mnih et al., 2015] incorporates a deep neural network, parameterized by θ, as a function approximator for the action-value function. The neural network parameters are estimated by minimizing the squared temporal difference residual:

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_\theta(s,a) - y\right)^2\right], \quad y = r + \gamma \max_{a\in\mathcal{A}} Q_{\theta^-}(s',a) \tag{2.5.3}$$

where transitions $(s, a, r = r(s,a), s' \sim P(s' \mid s,a))$ are sampled from a replay buffer of recent observed transitions $\mathcal{D}$. Here θ^- denotes the parameters of a target network which is updated ($\theta^- \leftarrow \theta$) regularly and held fixed between individual updates of θ.

2.5.2 Policy Gradient

Policy gradients methods search over a parameterized class of polices by performing stochastic gradient ascent on an objective function capturing the cumulative expected rewards. Specifically, they consider the scalar objective function

$$J^\pi \triangleq \mathbb{E}_{s\sim\rho}[V^\pi(s)] \tag{2.5.4}$$

which averages the total value function V^π over a random initial state distribution ρ.

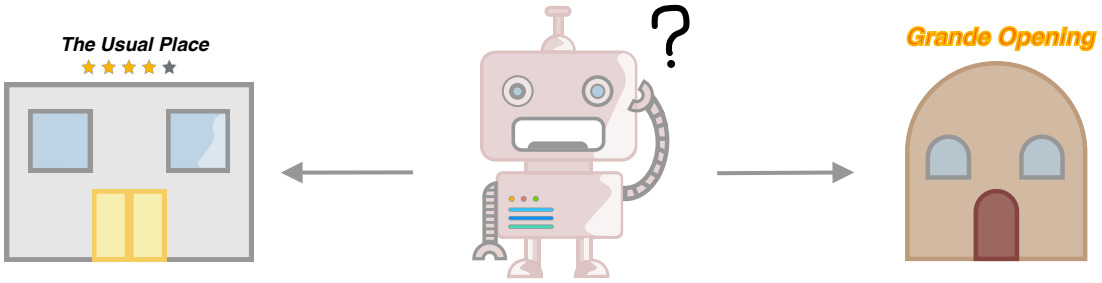

Figure 2.1: Restaurant selection example. Exploitation: Go to favorite restaurant. Exploration: Try a new restaurant

Let's first define the discounted state visitation distribution $d_\rho^{\pi_\theta}$ of policy π:

$$d_\rho^\pi(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr(s_t = s|s_0 \sim \rho, \pi) \tag{2.5.5}$$

where $\Pr(s_t = s|s_0 \sim \rho, \pi)$ is the state visitation probability that $s_t = s$, after we execute π starting at a state s_0 sampled from ρ.

For the class of stochastic differentiable policies $\{\pi_\theta, \theta \in \Theta \subset \mathbb{R}^d\}$, the fundamental result underlying these algorithms is the policy gradient theorem [Sutton et al., 1999a]:

$$\nabla_\theta J^{\pi_\theta} = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\rho^{\pi_\theta}, a\sim\pi_\theta}\Big[\nabla_\theta(\log\pi_\theta(a|s))Q^{\pi_\theta}(s,a)\Big] \tag{2.5.6}$$

The latter expression suggests that we can estimate the gradient by approximating the action-value function. It can be done by TD(0) with function approximation. This gave rise to the actor-critic architecture [Konda and Tsitsiklis, 2000].

Another instance of policy-gradient is the REINFORCE algorithm [Williams, 1992] which, instead of a learned value function, uses the cumulative return averaged over rollouts generated by the policy π_θ.

2.6 Exploitation-Exploration Dilemma

A RL agent aims at maximizing its long-term utility by *exploiting* its knowledge about the environment and allocating resources to the choices that are identified to be highly rewarding. However, this knowledge has to be acquired by the agent itself through *exploring* different choices that may reduce short-term utility. An illustrative situation is depicted in figure 2.1. An agent has the choice between going to its favorite restaurant so far or trying a new restaurant. A conservative agent would stick to its usual place that may be suboptimal choice. An active learner would experiment the new restaurant to figure out if it is worth

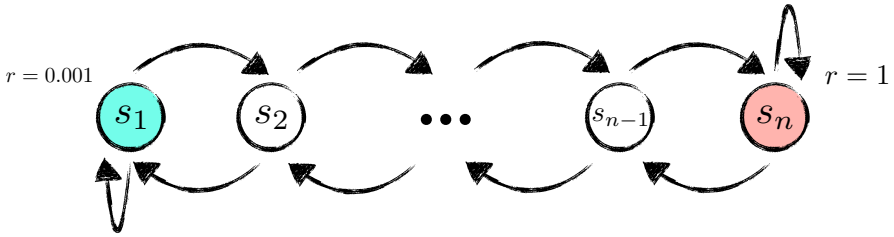

Figure 2.2: n-chain environment

going back to it and gain more knowledge. An efficient learning should balance carefully between exploration and exploitation.

The simplest approach to exploration is the ϵ-greedy strategy, which selects the greedy action with probability $1 - \epsilon$ and otherwise selects uniformly at random from all currently available actions. It is commonly used because it is easy to implement and disentangles explicitly between exploration and exploitation. However, such an approach does not take into account the level of information that actions could bring, and thus wastes exploratory resources.

To illustrate the inefficiency of random exploration, let us consider the n-chain environment in Figure 2.2. The environment consists of n states. The agent always starts at the second state s_2 and has two possible actions: move right or move left. A small reward $r = 0.001$ is received in the first state s_1, a large reward $r = 1$ in the final state s_n, otherwise the reward is zero. Q-learning with ϵ-greedy does poorly in this example. It takes $O(2^n)$ samples in expectation to visit the right end state for one time, resulting in an exponential sample complexity. What makes exploration particularly challenging in this toy example is that the agent only observes the informative signal after committing to the action right over an extended period of time. Therefore a *directed* exploration, involving a planning over time, is critical for effective learning.

Optimism in the face of uncertainty (OFU) is one of the traditional guiding principles that offers provably efficient learning algorithms. We can distinguish two classes of approaches: confidence-intervals based methods [Kearns and Singh, 2002, Strehl and Littman, 2005, Jaksch et al., 2010] and exploration-bonus based methods [Azar et al., 2017, Jin et al., 2018, Jian et al., 2019]. In the former, the agent builds a set of statistically plausible Markov Decision Processes (MDPs) that contains the true MDP with high probability. Then, the agent selects the most optimistic version of its model and acts optimally with respect to it. In the latter, discoveries of poorly understood states and actions are rewarded by an exploration bonus. Such bonus is designed to bound estimation errors on the value function.

An entire body of algorithms for efficient exploration is inspired by *Thompson sampling* [Thompson, 1933]. *Bayesian dynamic programming* was first introduced

in Strens [2000] and is more recently known as *posterior sampling for reinforcement learning* (PSRL) [Osband et al., 2013]. In PSRL, the agent starts with a prior belief over world models and then proceeds to update its full posterior distribution over models with the newly observed samples. A model hypothesis is then sampled from this distribution, and a greedy policy with respect to the sampled model is followed thereafter.

Why, intuitively, does PSRL works? let H_k the history of observations up to episode k. The posterior distribution $f(.|H_k)$ generates only plausible models and it concentrates around the True MDP.

- In early iterations, the posterior distribution is flat and its spread capture the uncertainty which enables exploration.
- As we collect more data, the mode of the posterior distribution captures the maximum likelihood estimates, which enable exploitation.

2.6.1 Online Performance

In order to quantitatively assess the exploration-exploitation trade-off, we use some online performance measures. the most common performance measures widely used in the literature are the *sample complexity of exploration* [Kakade, 2003] (defined for discounted and episodic MDPs) and the *regret* (defined traditionally for the average-reward and episodic MDPs and extended recently to discounted MDPs [Liu and Su, 2020])

In episodic MDPs, if an algorithm follows policy π_k in episode k, then the optimality gap in episode k is $\Delta_k \triangleq V_1^*(s_1) - V_1^{\pi_k}(s_1)$, where V_1^* and $V_1^{\pi_k}$ is respectively the optimal value function and the value function of π_k at stage $h = 1$, and s_1 is the initial state.

The sample complexity of exploration is defined as $N_(\epsilon, K) \triangleq \sum_{k=1}^{K} \mathbb{1}_{\{\Delta_k > \epsilon\}}$ i.e the total number of errors (ϵ-suboptimal policies). We say that a learning algorithm is (ϵ, δ)-PAC (Probably Approximately Correct) if there exists a polynomial function of $1/\epsilon$, δ and MDP's parameters such that $N_(\epsilon, K)$ is smaller than this function with probability at least $1 - \delta$. In other term, the number of errors can be bounded by a number polynomial in the relevant quantities with high probability.

The regret is defined as the total sum of optimality gap over episodes: $\text{REGRET}(K) = \sum_{k=1}^{K} \Delta_k$. A learning algorithm is said "efficient" if it achieves a sublinear regret i.e $\text{REGRET}(K) = o(K)$ when $K \leftarrow \infty$ in expectation or with high probability. A high probability bound is a stronger results as it can always be converted to a bound in expectation. Another quantity is the *expected Bayesian regret* where we consider the expectation of the regret over a prior distribution on MDPs. This is a weaker guarantee as the measure holds only in expectation over MDPs and not for a specific MDP.

The tow aforementioned performance measures are not directly comparable. An algorithm with small regret can make small mistakes infinitely often and thus it cannot be (ϵ, δ)-PAC for ϵ small enough. Dann et al. [2017] provide an extensive study of the connection between the two quantities.

3 Multi-step Off-policy Learning with Function Approximation

«Before a decision is made, a set of options constitute candidates for future reality. After the decision has been made and its consequences are known, those unchosen options can never become reality, but they may nevertheless haunt us, amuse us, or influence our perception of the decision process itself.»
– [Roese, 1999]

3.1 Prologue to the Contribution

3.1.1 Article Details

This chapter is based on the article «*Convergent* Tree-backup *and* Retrace *with Function Approximation*» [Touati et al., 2018]. This is joint work with Pierre-Luc Bacon, Doina Precup and Pascal Vincent. This paper was accepted as a long oral at ICML 2018. I am the main author. I performed the theoretical analysis, the derivation of the new algorithm and the experiments.

3.1.2 Context

This work started originally as a course project in the context of the RL class taught by Doina Precup at Mcgill university. I was interested in optimization-based RL algorithms. Pierre-luc Bacon, who was the teaching assistant at the time, suggested to me to study the behavior of Retrace algorithm with function approximation. Retrace was recently introduced as a safe and efficient off-policy method in Munos et al. [2016] and became quickly at the heart of several successful deep RL architectures such as Reactor (Retrace-Actor) [Gruslys et al., 2017] and ACER [Wang et al., 2016].

3.1.3 Paper Abstract

Off-policy learning is key to scaling up reinforcement learning as it allows to learn about a target policy from the experience generated by a different behavior policy. Unfortunately, it has been challenging to combine off-policy learning

with function approximation and multi-step bootstrapping in a way that leads to both stable and efficient algorithms.

In this work, we show that popular multi-step off-policy methods such as TREE BACKUP [Precup, 2000] and RETRACE [Munos et al., 2016] are unstable with linear function approximation, both in theory and in practice with specific examples. Based on our analysis, we then derive stable and efficient gradient-based algorithms using a quadratic convex-concave saddle-point formulation. By exploiting the problem structure proper to these algorithms, we are able to provide convergence guarantees and finite-sample bounds. The applicability of our new analysis also goes beyond TREE BACKUP and RETRACE and allows us to provide new state-of-the-art convergence rates for the GTD and GTD2 algorithms without having recourse to projections or Polyak averaging.

3.1.4 Recent Developments

While our work focuses on linear function approximation, Wai et al. [2019] extend the primal-dual formulation of policy evaluation to nonlinear smooth function approximation, show that it converges to a stationary point and provide its finite sample analysis.

I worked on a followup in [Peng et al., 2019], jointly led by Zilun Peng and in collaboration with Pascal Vincent and Doina Precup. We proposed a reduced variance methods for the problem of off-policy policy evaluation that achieve a linear convergence rate while addressing the computational bottleneck that previously proposed reduced variance methods in Du et al. [2017] suffer from.

3.2 INTRODUCTION

Rather than being confined to their own stream of experience, off-policy learning algorithms are capable of leveraging data from a different behavior than the one being followed, which can provide many benefits: efficient parallel exploration as in Mnih et al. [2016] and Wang et al. [2016], reuse of past experience with experience replay [Lin, 1992] and, in many practical contexts, learning form data produced by policies that are currently deployed, but which we want to improve (as in many scenarios of working with an industrial or health care partner). Moreover, a single stream of experience can be used to learn about a variety of different targets which may take the form of value functions corresponding to different policies and time scales [Sutton et al., 1999c] or to predicting different reward functions as in Sutton and Tanner [2004] and Sutton et al. [2011]. Therefore, the design and analysis of off-policy algorithms using all the features of reinforcement learning, e.g. bootstrapping, multi-step updates (eligibility traces), and function approximation has been explored extensively over

three decades. While off-policy learning and function approximation have been understood in isolation, their combination with multi-steps bootstrapping produces a so-called *deadly triad* [Sutton, 2015, Sutton and Barto, 2018], i.e., many algorithms in this category are unstable.

A typical off-policy approach to is provided by importance sampling, which bends the behavior policy distribution onto the target one [Precup, 2000, Precup et al., 2001]. However, as the length of the trajectories increases, the variance of importance sampling corrections tends to become very large. The TREE BACKUP algorithm [Precup, 2000] is an alternative approach which remarkably does not rely on importance sampling ratios directly. More recently, Munos et al. [2016] introduced the RETRACE algorithm which also builds on TREE BACKUP to perform off-policy learning without importance sampling.

Until now, TREE BACKUP and RETRACE(λ) had only been shown to converge in the tabular case, and their behavior with linear function approximation was not known. In this paper, we show that this combination with linear function approximation is in fact divergent. We obtain this result by analyzing the mean behavior of TREE BACKUP and RETRACE using the ordinary differential equation (ODE) [Borkar and Meyn, 2000] associated with them. We also demonstrate this instability with a concrete counterexample.

Insights gained from this analysis allow us to derive a new gradient-based algorithm with provable convergence guarantees. Instead of adapting the derivation of Gradient Temporal Difference (GTD) learning from [Sutton et al., 2009c], we use a primal-dual saddle point formulation [Liu et al., 2015, Macua et al., 2015] which facilitates the derivation of sample complexity bounds. The underlying saddle-point problem combines the primal variables, function approximation parameters, and dual variables through a bilinear term.

In general, stochastic primal-dual gradient algorithms like the ones derived in this paper can be shown to achieve $O(1/k)$ convergence rate (where k is the number of iterations). For example, this has been established for the class of forward-backward algorithms with added noise [Rosasco et al., 2016]. Furthermore, this work assumes that the objective function is composed of a convex-concave term and a strongly convex-concave regularization term that admits a tractable proximal mapping. In this paper, we are able to achieve the same $O(1/k)$ convergence rate without having to assume strong convexity with respect to the primal variables and in the absence of proximal mappings. As corollary, our convergence rate result extends to the well-known gradient-based temporal difference algorithms GTD [Sutton et al., 2009c] and GTD2 [Sutton et al., 2009b] and hence improves the previously published results.

The algorithms resulting from our analysis are simple to implement, and perform well in practice compared to other existing multi-steps off-policy learning algorithms such as GQ(λ) [Maei and Sutton, 2010] and AB-TRACE(λ) [Mahmood et al., 2017].

3.3 TABULAR OFF-POLICY METHODS

In this work, we are concerned with the policy evaluation problem [Sutton and Barto, 1998] under model-free off-policy learning. That is, we will evaluate a *target* policy π using trajectories (i.e. sequences of states, actions and rewards) obtained from a different *behavior* policy μ.

Munos et al. [2016] provided a unified perspective on several off-policy learning algorithms, namely: those using explicit importance sampling corrections [Precup, 2000] as well as TREE BACKUP (TB(λ)) [Precup, 2000] and $Q^\pi(\lambda)$ [Harutyunyan et al., 2016] which do not involve importance ratios. As a matter of fact, all these methods share a general form based on the λ-return [Sutton and Barto, 2018] but involve different coefficients κ_i in :

$$\begin{aligned} G_k^\lambda &\triangleq Q(s_k, a_k) + \sum_{t=k}^{\infty} (\lambda\gamma)^{t-k} \left(\prod_{i=k+1}^{t} \kappa_i \right) \left(r_t + \gamma \mathbb{E}_\pi Q(s_{t+1}, \cdot) - Q(s_t, a_t) \right) \\ &= Q(s_k, a_k) + \sum_{t=k}^{\infty} (\lambda\gamma)^{t-k} \left(\prod_{i=k+1}^{t} \kappa_i \right) \delta_t \ , \end{aligned}$$

where $\mathbb{E}_\pi Q(s_{t+1}, .) \triangleq \sum_{a \in \mathcal{A}} \pi(a \mid s_{t+1}) Q(s_{t+1}, a)$ and $\delta_t \triangleq r_t + \gamma \mathbb{E}_\pi Q(s_{t+1}, .) - Q(s_t, a_t)$ is the temporal-difference (TD) error. The coefficients κ_i determine how the TD errors would be scaled in order to correct for the discrepancy between target and behavior policies. From this unified representation, Munos et al. [2016] derived the RETRACE(λ) algorithm. Both TB(λ) and RETRACE(λ) consider this form of return, but set κ_i differently. The TB(λ) updates correspond to the choice $\kappa_i = \pi(a_i \mid s_i)$ while RETRACE(λ) sets $\kappa_i = \min\left(1, \frac{\pi(a_i|s_i)}{\mu(a_i|s_i)}\right)$, which is intended to allow learning from full returns when the target and behavior policies are very close. The importance sampling approach [Precup, 2000] converges in the tabular case by correcting the behavior data distribution to the distribution that would be induced by the target policy π ($\kappa_i = \frac{\pi(a_i|s_i)}{\mu(a_i|s_i)}$). However, these correction terms lead to high variance in practice. Since $Q(\lambda)$ does not involve importance ratios ($\kappa_i = 1$), this variance problem is avoided but at the cost of restricted convergence guarantees satisfied only when the behavior and target policies are sufficiently close.

The analysis provided in this paper concerns TB(λ) and RETRACE(λ), which are convergent in the tabular case, but have not been analyzed in the function approximation case. We start by noting that the Bellman operator [1] $\mathcal{R}$ underly-

[1] We overload our notation over linear operators and their corresponding matrix representation.

	Value of κ_i	l_∞ contraction guarantee	Estimation variance
Importance sampling	$\frac{\pi(a_i\|s_i)}{\mu(a_i\|s_i)}$	for any π, μ	High
$Q^\pi(\lambda)$	1	only for π close to μ	Low
TB(λ)	$\pi(a_i \mid s_i)$	for any π, μ	Low
RETRACE(λ)	$\min\left(1, \frac{\pi(a_i\|s_i)}{\mu(a_i\|s_i)}\right)$	for any π, μ	Low

Table 3.1: Properties of different off-policy algorithms for policy evaluation.

ing these algorithms can be written in the following form:

$$(\mathcal{R}Q)(s,a) \triangleq Q(s,a) + \mathbb{E}_\mu\Big[\sum_{t=0}^{\infty}(\lambda\gamma)^t\left(\prod_{i=1}^{t}\kappa_i\right)(r_t + \gamma\mathbb{E}_\pi Q(s_{t+1},\cdot) - Q(s_t,a_t))\Big]$$
$$= Q(s,a) + (I - \lambda\gamma P^{\kappa\mu})^{-1}(\mathcal{T}^\pi Q - Q)(s,a) \ ,$$

where $\mathbb{E}_\mu$ is the expectation over the behavior policy and MDP transition probabilities and $P^{\kappa\mu}$ is the operator defined by:

$$(P^{\kappa\mu}Q)(s,a) \triangleq \sum_{\substack{s'\in\mathcal{S}\\ a'\in\mathcal{A}}} P(s' \mid s,a)\mu(a' \mid s')\kappa(s',a')Q(s',a') \ .$$

In the tabular case, these operators were shown to be contraction mappings with respect to the supremum norm [Precup, 2000, Munos et al., 2016]. In this paper, we focus on what happens to these operators when combined with linear function approximation.

3.4 OFF-POLICY INSTABILITY WITH FUNCTION APPROXIMATION

In order to obtain generalization between different state-action pairs, Q^π should be represented in a functional form. In this paper, we focus on linear function approximation of the form:

$$Q(s,a) \triangleq \theta^\top\phi(s,a) \ ,$$

where $\theta \in \Theta \subset \mathbb{R}^d$ is a weight vector and $\phi : \mathcal{S}\times\mathcal{A} \to \mathbb{R}^d$ is a feature map from a state-action pairs to a given d-dimensional feature space.

When combined with function approximation, the temporal difference updates corresponding to the λ-return G_k^λ are given by

$$\begin{aligned}\theta_{k+1} &= \theta_k + \alpha_k \left(G_k^\lambda - Q(s_k, a_k)\right) \nabla_\theta Q(s_k, a_k) \\ &= \theta_k + \alpha_k \left(\sum_{t=k}^{\infty} (\lambda\gamma)^{t-k} \left(\prod_{i=k+1}^{t} \kappa_i\right) \delta_t^k\right) \phi(s_k, a_k) \end{aligned} \tag{3.4.1}$$

where $\delta_t^k = r_t + \gamma\theta_k^\top \mathbb{E}_\pi \phi(s_{t+1}, \cdot) - \theta_k^\top \phi(s_t, a_t)$ and α_k are positive non-increasing step sizes. The updates (3.4.1) implies off-line updating as G_k^λ is a quantity which depends on future rewards. This will be addressed later using eligibility traces: a mechanism to transform the off-line updates into efficient on-line ones. Since (3.4.1) describes stochastic updates, the following standard assumption is necessary:

Assumption 1. *The Markov chain induced by the behavior policy μ is ergodic and admits a unique stationary distribution, denoted by ξ, over state-action pairs. We write Ξ for the diagonal matrix whose diagonal entries are $(\xi(s,a))_{s\in\mathcal{S}, a\in\mathcal{A}}$.*

Our first proposition establishes the expected behavior of the parameters in the limit.

Proposition 1. *If the behavior policy satisfies Assumption 1 and $(\theta_k)_{k\leq 0}$ is the Markov process defined by (3.4.1) then:*

$$\mathbb{E}[\theta_{k+1} \mid \theta_0] = (I + \alpha_k A)\, \mathbb{E}[\theta_k \mid \theta_0] + \alpha_k b \ ,$$

where matrix A and vector b are defined as follows:

$$\begin{aligned} A &\triangleq \Phi^\top \Xi (I - \lambda\gamma P^{\kappa\mu})^{-1} (\gamma P^\pi - I)\Phi \ , \\ b &\triangleq \Phi^\top \Xi (I - \lambda\gamma P^{\kappa\mu})^{-1} r \ . \end{aligned}$$

Sketch of Proof (The full proof is in the appendix).

$$\begin{aligned}\theta_{k+1} &= \theta_k + \alpha_k \left(\sum_{t=k}^{\infty} (\lambda\gamma)^{t-k} \left(\prod_{i=k+1}^{t} \kappa_i\right) \phi(s_k, a_k)([\gamma \mathbb{E}_\pi \phi(x_{t+1}, \cdot) - \phi(x_t, a_t)]^\top \theta_k + r_t)\right) \\ &= \theta_k + \alpha_k \left(A_k \theta_k + b_k\right) \ . \end{aligned}$$

So, $\mathbb{E}[\theta_{k+1} \mid \theta_k] = (I + \alpha_k A)\theta_k + \alpha_k b$ where $A = \mathbb{E}[A_k]$ and $b = \mathbb{E}[b_k]$ □

The ODE (Ordinary Differential Equations) approach [Borkar and Meyn, 2000] is the main tool to establish convergence in the function approximation case [Bertsekas and Tsitsiklis, 1995, Tsitsiklis et al., 1997]. In particular, we use Proposition 4.8 in Bertsekas and Tsitsiklis [1995], which states that under some

conditions, θ_k converges to the unique solution θ^* of the system $A\theta^* + b = 0$. This crucially relies on the matrix A being negative definite i.e $y^\top Ay < 0, \forall y \neq 0$. In the on-policy case, when $\mu = \pi$, we rely on the fact that the stationary distribution is invariant under the transition matrix P^π i.e $d^\top P^\pi = d^\top$ [Tsitsiklis et al., 1997, Sutton et al., 2015]. However, this is no longer true for off-policy learning with arbitrary target/behavior policies and the matrix A may not be negative definite: the series θ_k may then diverge. We will now see that the same phenomenon may occur with TB(λ) and RETRACE(λ).

Counterexample: We extend the two-states MDP of Tsitsiklis et al. [1997], originally proposed to show the divergence of off-policy TD(0), to the case of function approximation over state-action pairs. This environment has only two states, as shown in Figure 3.1, and two actions: left or right.

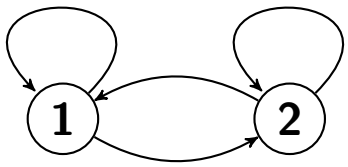

Figure 3.1: Two-state counterexample. We assign the features $\{(1,0)^\top, (2,0)^\top, (0,1)^\top, (0,2)^\top\}$ to the state-action pairs $\{(1, \text{right}), (2, \text{right}), (1, \text{left}), (2, \text{left})\}$. The target policy is given by $\pi(\text{right} \mid \cdot) = 1$ and the behavior policy is $\mu(\text{right} \mid \cdot) = 0.5$

In this particular case, both TB(λ) and RETRACE(λ) share the same matrix $P^{\kappa\mu}$ and $P^{\kappa\mu} = 0.5P^\pi$:

$$P^\pi = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, (P^\pi)^n = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \forall n \geq 2$$

If we set $\beta := 0.5\gamma\lambda$, we then have:

$$(I - \lambda\gamma P^{\kappa\mu})^{-1} = \begin{pmatrix} 1 & \frac{\beta}{1-\beta} & 0 & 0 \\ 0 & \frac{1}{1-\beta} & 0 & 0 \\ \beta & \frac{\beta^2}{1-\beta} & 1 & 0 \\ \beta & \frac{\beta^2}{1-\beta} & 0 & 1 \end{pmatrix}, A = \begin{pmatrix} \frac{6\gamma-\beta-5}{1-\beta} & 0 \\ \frac{3(\gamma\beta-\beta^2-\beta-\gamma)}{1-\beta} & -5 \end{pmatrix} .$$

Therefore, $\forall\, \gamma \in (\frac{5}{6}, 1)$ and $\forall\, \lambda \in [0, \min(1, \frac{12\gamma-10}{\gamma}))$, the first eigenvalue $e_1 = \frac{6\gamma-\beta-5}{1-\beta}$ of A is positive. The basis vectors $(1,0)^\top$ and $(0,1)^\top$ are eigenvectors of A associated with e_1 and -5, then if $\theta_0 = (\eta_1, \eta_2)^\top$, we obtain

$\mathbb{E}[\theta_k \mid \theta_0] = (\eta_1 \prod_{i=0}^{k-1}(1+\alpha_i e_1), \eta_2 \prod_{i=0}^{k-1}(1-5\alpha_i))^\top$ implying that $||\mathbb{E}[\theta_k \mid \theta_0]|| \geq |\eta_1| \prod_{i=0}^{k-1}(1+\alpha_i e_1)$. Hence, as $\sum_k \alpha_k \to \infty$, $||\mathbb{E}[\theta_k \mid \theta_0]|| \to \infty$ if $\eta_1 \neq 0$.

3.5 CONVERGENT GRADIENT OFF-POLICY ALGORITHMS

If A were to be negative definite, RETRACE(λ) or TB(λ) with function approximation would converge to $\theta^* = -A^{-1}b$. It is known [Bertsekas, 2011] that $\Phi\theta^*$ is the fixed point of the projected Bellman operator :

$$\Phi\theta^* = \Pi^\mu \mathcal{R}(\Phi\theta^*) \ ,$$

where $\Pi^\mu = \Phi(\Phi^\top \Xi \Phi)^{-1}\Phi^\top \Xi$ is the orthogonal projection onto the space $S = \{\Phi\theta | \theta \in \mathbb{R}^d\}$ with respect to the weighted Euclidean norm $||.||_\Xi$. Rather than computing the sequence of iterates given by the projected Bellman operator, another approach for finding θ^* is to directly minimize [Sutton et al., 2009a, Liu et al., 2015] the Mean Squared Projected Bellman Error (MSPBE):

$$\textbf{MSPBE}(\theta) = \frac{1}{2}||\Pi^\mu \mathcal{R}(\Phi\theta) - \Phi\theta||^2_\Xi \ .$$

This is the route that we take in this paper to derive convergent forms of TB(λ) and RETRACE(λ). To do so, we first define our objective function in terms of A and b which we introduced in Proposition 1.

Proposition 2. *Let $M \triangleq \Phi^\top \Xi \Phi = \mathbb{E}[\Phi\Phi^\top]$ be the covariance matrix of features. We have:*

$$\textbf{MSPBE}(\theta) = \frac{1}{2}||A\theta + b||^2_{M^{-1}}$$

(The proof is provided in the appendix.)

In order to derive parameter updates, we could compute gradients of the above expression explicitly as in Sutton et al. [2009c], but we would then obtain a gradient that is a product of expectations. The implied double sampling makes it difficult to obtain an unbiased estimator of the gradient. Sutton et al. [2009c] addressed this problem with a two-timescale stochastic approximations. However, the algorithm obtained in this way is no longer a true stochastic gradient method with respect to the original objective. Liu et al. [2015] suggested an alternative which converts the original minimization problem into a primal-dual saddle-point problem. This is the approach that we chose in this paper.

The convex conjugate of a real-valued function f is defined as:

$$f^*(y) = \sup_{x \in X}(\langle y, x \rangle - f(x)) \ , \tag{3.5.1}$$

and f is convex, we have $f^{**} = f$. Also, if $f(x) = \frac{1}{2}||x||_{M^{-1}}$, then $f^*(x) = \frac{1}{2}||x||_M$. Note that by going to the convex conjugate, we do not need to invert matrix M. We now go back to the original minimization problem:

$$\begin{aligned}\min_\theta \mathbf{MSPBE}(\theta) &\Leftrightarrow \min_\theta \frac{1}{2}||A\theta + b||^2_{M^{-1}} \\ &\Leftrightarrow \min_\theta \max_\omega \left(\langle A\theta + b, \omega \rangle - \frac{1}{2}||\omega||^2_M \right)\end{aligned}$$

The gradient updates resulting from the saddle-point problem (ascent in ω and descent in θ) are then:

$$\begin{aligned}\omega_{k+1} &= \omega_k + \eta_k(A\theta_k + b - M\omega_k) \ , \\ \theta_{k+1} &= \theta_k - \alpha_k A^\top \omega_k \ .\end{aligned} \tag{3.5.2}$$

where $\{\eta_k\}$ and $\{\alpha_k\}$ are non-negative step-size sequences. As the A, b and M are all expectations, we can derive stochastic updates by drawing samples, which would yield unbiased estimates of the gradient.

On-line updates: We now derive on-line updates by exploiting equivalences in expectation between forward views and backward views outlined in Maei [2011].

Proposition 3. *Let e_k be the eligibility traces vector, defined as $e_{-1} = 0$ and :*

$$e_k = \lambda\gamma\kappa(s_k, a_k)e_{k-1} + \phi(s_k, a_k) \quad \forall k \geq 0 \ .$$

Furthermore, let $\hat{A}_k = e_k(\gamma\mathbb{E}_\pi[\phi(s_{k+1}, .)] - \phi(s_k, a_k)])^\top$, $\hat{b}_k = r(s_k, a_k)e_k$, $\hat{M}_k = \phi(s_k, a_k)\phi(s_k, a_k)^\top$. Then, we have $\mathbb{E}[\hat{A}_k] = A$, $\mathbb{E}[\hat{b}_k] = b$ and $\mathbb{E}[\hat{M}_k] = M$.
(The proof is provided in the appendix.)

This proposition allows us to replace the expectations in Eq. (3.5.2) by corresponding unbiased estimates. The resulting detailed procedure is provided in Algorithm 3.

3.6 Convergence Rate Analysis

In order to characterize the convergence rate of the algorithm 3, we need to introduce some new notations and state new assumptions.

We denote by $\|A\| \triangleq \sup_{\|x\|=1} \|Ax\|$ the spectral norm of the matrix A and by $c(A) = \|A\|\|A^{-1}\|$ its condition number. If the eigenvalues of a matrix A

Algorithm 3 Gradient Off-policy with eligibility traces

Given: target policy π, behavior policy μ
  Initialize θ_0 and ω_0
  for n = 0 ... **do**
    set $e_0 = 0$
    for k = 0 ...end of episode **do**
      Observe s_k, a_k, r_k, s_{k+1} according to μ
      /* Update traces
      $e_k = \lambda\gamma\kappa(s_k, a_k)e_{k-1} + \phi(s_k, a_k)$
      /* Update parameters
      $\delta_k = r_k + \gamma\theta_k^\top \mathbb{E}_\pi\phi(s_{k+1}, .) - \theta_k^\top\phi(s_k, a_k)$
      $\omega_{k+1} = \omega_k + \eta_k \left(\delta_k e_k - \omega_k^\top\phi(s_k, a_k)\phi(s_k, a_k)\right)$
      $\theta_{k+1} = \theta_k - \alpha_k\omega_k^\top e_k \left(\gamma\mathbb{E}_\pi\phi(s_{k+1}, .) - \phi(s_k, a_k)\right)$
    end for
  end for

are real, we use $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ to denote respectively the largest and the smallest eigenvalue.

If we set $\eta_k = \beta\alpha_k$ for a positive constant β, it is possible to combine the two iterations present in our algorithm as a single iteration using a parameter vector $z_k \triangleq \begin{pmatrix} \theta_k \\ \frac{1}{\sqrt{\beta}}\omega_k \end{pmatrix}$ where :

$$z_{k+1} = z_k - \alpha_k(\hat{G}_k z_k - \hat{g}_k)$$

where:

$$\hat{G}_k \triangleq \begin{pmatrix} 0 & \sqrt{\beta}\hat{A}_k^\top \\ -\sqrt{\beta}\hat{A}_k & \beta\hat{M}_k \end{pmatrix} \quad \hat{g}_k \triangleq \begin{pmatrix} 0 \\ \sqrt{\beta}\hat{b}_k \end{pmatrix}$$

Let $G \triangleq \mathbb{E}\left[\hat{G}_k\right]$ and $g = \mathbb{E}\left[\hat{g}_k\right]$. It follows from the proposition 3 that G and g are well defined and more specifically:

$$G = \begin{pmatrix} 0 & \sqrt{\beta}A^\top \\ -\sqrt{\beta}A & \beta M \end{pmatrix} \quad g = \begin{pmatrix} 0 \\ \sqrt{\beta}b \end{pmatrix}$$

Furthermore, let $\mathcal{F}_k = \sigma(z_0, \hat{G}_0, \hat{g}_0 \ldots, z_k, \hat{G}_k, \hat{g}_k, z_{k+1})$ be the sigma-algebra generated by the variables up to time k. With these definitions, we can now state our assumptions.

Assumption 2. *The matrices A and M are nonsingular. This implies that the saddle-point problem admits a unique solution $(\theta^\star, \omega^\star) = (-A^{-1}b, 0)$ and we define $z^\star \triangleq (\theta^\star, \frac{1}{\sqrt{\beta}}\omega^\star)$.*

Assumption 3. *The features and reward functions are uniformly bounded. This implies that the features and rewards have uniformly bounded second moments. It follows that there exists a constant σ such that:*

$$\mathbb{E}[\|\hat{G}_k z_k - \hat{g}_k\|^2 | \mathcal{F}_{k-1}] \leq \sigma^2(1 + \|z_k\|^2)$$

Before stating our main result, the following key quantities needs to be defined:

$$\rho \triangleq \lambda_{\max}(A^\top M^{-1} A), \quad \delta \triangleq \lambda_{\min}(A^\top M^{-1} A),$$
$$L_G \triangleq \left\| \mathbb{E}\left[\hat{G}_k^\top \hat{G}_k \mid \mathcal{F}_{k-1} \right] \right\|$$

The following proposition characterize the convergence in expectation of $\|z_k - z^\star\|^2 = \|\theta_k - \theta^\star\|^2 + \frac{1}{\beta}\|w_k\|^2$

Proposition 4. *Suppose assumptions 2 and 3 holds and if we choose $\beta = \frac{8\rho}{\lambda_{\min}(M)}$ and $\alpha_k = \frac{9^2 \times 2\delta}{8\delta^2(k+2)+9^2\zeta}$ where $\zeta = 2 \times 9^2 c(M)^2\rho^2 + 32c(M)L_G$. Then the mean square error $\mathbb{E}\left[\|z_k - z^\star\|^2\right]$ is upper bounded by:*

$$9^2 \times 8c(M)\left\{ \frac{(8\delta + 9\zeta)^2 \mathbb{E}\left[\|z_0 - z^\star\|^2\right]}{(8^2\delta^2 k + 9^2\zeta)^2} + \frac{8\sigma^2(1 + \|z^\star\|^2)}{(8^2\delta^2 k + 9^2\zeta)} \right\}$$

Sketch of Proof (The full proof is in the appendix). The beginning of our proof relies on Du et al. [2017] which shows the linear convergence rate of deterministic primal-dual gradient method for policy evaluation. More precisely, we make use of the spectral properties of matrix G shown in the appendix of this paper. The rest of the proof follows a different route exploiting the structure of our problem. □

The above proposition 4 shows that the mean square error $\mathbb{E}\left[\|z_k - z^\star\|^2\right]$ at iteration k is upper bounded by tow terms. The first bias term tells that the initial error $\mathbb{E}\left[\|z_0 - z^\star\|^2\right]$ is forgotten at a rate $O(1/k^2)$ and the constant depends on the condition number of the covariance matrix $c(M)$. The second variance term shows that noise is rejected at a rate $O(1/k)$ and the constant depends on the variance of estimates σ^2 and $c(M)$. The overall convergence rate is $O(1/k)$.

Existing stochastic saddle-point problem results: Chen et al. [2014] provides a comprehensive review of stochastic saddle-point problem. When the objective function is convex-concave, the overall convergence rate is $O(1/\sqrt{k})$. Although several accelerated techniques could improve the dependencies on the smoothness constants of the problem in their convergence rate, the dominant term that depends on the gradient variance still decays only as $O(1/\sqrt{k})$.

Paper	step-sizes	Projection	Polyak averaging	Convergence rate
Sutton et al. [2009c], Sutton et al. [2009b]	$\eta_k = \beta\alpha_k, \beta > 0,$ $\sum_k \alpha_k = \infty, \sum_k \alpha_k^2 < \infty$	No	No	$\theta_k \to \theta^\star$ with probability one
Liu et al. [2015]	constant step-size, $\alpha_k = \eta_k$	Yes	Yes	$\mathbf{MSPBE}(\bar{\theta}_k) \in O(1/\sqrt{k})$ with high probability
Wang et al. [2017]	$\alpha_k = \eta_k, \sum_k \alpha_k = \infty,$ $\frac{\sum_k \alpha_k^2}{\sum_k \alpha_k} < \infty$	Yes	Yes	$\mathbf{MSPBE}(\bar{\theta}_k) \in O(\frac{\sum_k \alpha_k^2}{\sum_k \alpha_k})$ with high probability
Lakshminarayanan and Szepesvári [2017]	constant step-size, $\alpha_k = \eta_k$	No	Yes	$\mathbb{E}[\|\bar{\theta}_k - \theta^\star\|^2] \in O(1/k)$
Dalal et al. [2017]	$\alpha_k = \frac{1}{k^{1-c}},$ $\eta_k = \frac{1}{k^{(2/3)(1-c)}}$ where $c \in (0,1)$	Yes	No	$\|\theta_k - \theta^\star\| \in O(k^{-\frac{1}{3}+\frac{c}{3}})$ with high probability
Our work	$\eta_k = \beta\alpha_k, \beta > 0,$ $\alpha_k \in O(1/k)$	No	No	$\mathbb{E}[\|\theta_k - \theta^\star\|^2] \in O(1/k)$

Table 3.2: Convergence results for gradient-based TD algorithms shown in previous work [Sutton et al., 2009b,c, Liu et al., 2015, Wang et al., 2017, Lakshminarayanan and Szepesvári, 2017, Dalal et al., 2017]. $\bar{\theta}_k$ stand for the Polyak-average of iterates: $\bar{\theta}_k \triangleq \frac{\sum_k \alpha_k \theta_k}{\sum_k \alpha_k}$. Our algorithms achieve $O(1/k)$ without the need for projections or Polyak averaging.

When the objective function is strongly convex-concave, Rosasco et al. [2016] and Palaniappan and Bach [2016] showed that stochastic forward-backward algorithms can achieve $O(1/k)$ convergence rate. Algorithms in this class are feasible in practice only if their proximal mappings can be computed efficiently. In our case, our objective function is strongly concave because of the positive-definiteness of M but is otherwise not strongly convex. Because our algorithms are vanilla stochastic gradient methods, they do not rely on proximal mappings.

Singularity: If assumption 2 does not hold, the matrix G is singular and either $Gz + g = 0$ has infinitely many solutions or it has no solution. In the case of many solutions, we could still get asymptotic convergence. In Wang and Bertsekas [2013], it was shown that under some assumptions on the null space of matrix G and using a simple stabilization scheme, the iterates converge to the Drazin [Drazin, 1958] inverse solution of $Gz + g = 0$. However, it is not clear how extend our finite-sample analysis because the spectral analysis of the matrix G [Benzi and Simoncini, 2006] in our proof assumes that the matrices A and M are nonsingular.

3.7 Related Work and Discussion

Convergent Retrace: Mahmood et al. [2017] have recently introduced the ABQ(ζ) algorithm which uses an action-dependent bootstrapping parameter that leads to off-policy multi-step learning without importance sampling ratios. They also derived a gradient-based algorithm called AB-Trace(λ) which is related to Retrace(λ). However, the resulting updates are different from ours, as they use the two-timescale approach of Sutton et al. [2009a] as basis for their derivation. In contrast, our approach uses the saddle-point formulation, avoiding the need for double sampling. Another benefit of this formulation is that it allows us to provide a bound of the convergence rate (proposition 4) whereas Mahmood et al. [2017] is restricted to a more general two-timescale asymptotic result from Borkar and Meyn [2000]. The saddle-point formulation also provides a rich literature on acceleration methods which could be incorporated in our algorithms. Particularly in the batch setting, Du et al. [2017] recently introduced Stochastic Variance Reduction methods for state-value estimation combining GTD with SVRG Johnson and Zhang [2013] or SAGA Defazio et al. [2014]. This work could be extended easily to ours algorithms in the batch setting.

Existing Convergence Rates: Our convergence rate result 4 can apply to GTD/GTD2 algorithms. Recall that GTD/GTD2 are off-policy algorithms designed to estimate the state-value function using temporal difference TD(0) return while our algorithms compute the action-value function using Retrace and Tree Backup returns. In both GTD and GTD2, the quantities $\hat{A}_k$ and $\hat{b}_k$ involved in their updates are the same and equal to $\hat{A}_k = \phi(s_k)(\gamma\phi(s_{k+1}) - \phi(s_k))^\top$, $\hat{b}_k = r(s_k, a_k)\phi(s_k)$ while the matrix $\hat{M}_k$ is equal to $\phi(s_k)\phi(s_k)^\top$ for GTD2 and to identity matrix for GTD.
The table 3.2 show in chronological order the convergence rates established in the literature of Reinforcement learning. GTD was first introduced in Sutton et al. [2009c] and its variant GTD2 was introduced later in Sutton et al. [2009b]. Both papers established the asymptotic convergence with Robbins-Monro step-sizes. Later, Liu et al. [2015] provided the first sample complexity by reformulating GTD/GTD2 as an instance of mirror stochastic approximation [Nemirovski et al., 2009]. Liu et al. [2015] showed that with high probability, $\mathbf{MSPBE}(\bar{\theta}_k) \in O(1/\sqrt{k})$ where $\bar{\theta}_k \triangleq \frac{\sum_k \alpha_k \theta_k}{\sum_k \alpha_k}$. However, they studied an alternated version of GTD/GTD2 as they added a projection step into bounded convex set and Polyak-averaging of iterates. Wang et al. [2017] studied also the same version as Liu et al. [2015] but for the case of Markov noise case instead of the *i.i.d* assumptions. They prove that with high probability $\mathbf{MSPBE}(\bar{\theta}_k) \in O(\frac{\sum_k \alpha_k^2}{\sum_k \alpha_k})$ when the step-size sequence satisfies $\sum_k \alpha_k = \infty, \frac{\sum_k \alpha_k^2}{\sum_k \alpha_k} < \infty$. The optimal

rate achieved in this setup is then $O(1/\sqrt{k})$. Recently, Lakshminarayanan and Szepesvári [2017] improved on the existing results by showing for the first time that $\mathbb{E}[\|\bar{\theta}_k - \theta^\star\|^2] \in O(1/k)$ without projection step. However, the result still consider the Polyak-average of iterates. Moreover, the constants in their bound depend on the data distribution that are difficult to relate to the problem-specific constants, such as those present in our bound 4. Finally, Dalal et al. [2017] studied sparsily projected version of GTD/GTD2 and they showed that for step-sizes $\alpha_k = \frac{1}{k^{1-c}}, \eta_k = \frac{1}{k^{(2/3)(1-c)}}$ where $c \in (0,1)$, $\|\theta_k - \theta^\star\| \in O(k^{-\frac{1}{3}+\frac{c}{3}})$ with high probability. The projection is called sparse as they project only on iterations which are powers of 2.
Our work is the first to provide a finite-sample complexity analysis of GTD/GTD2 in its original setting, i.e without assumption a projection step or Polyak-averaging and with diminishing step-sizes.

3.8 Experimental Results

3.8.1 Evidence of instability in practice

To validate our theoretical results about instability, we implemented TB(λ), Retrace(λ) and compared them against their gradient-based counterparts GTB(λ) and GRetrace(λ) derived in this paper. The first one is the 2-states counterexample that we detailed in the third section and the second is the 7-states versions of Baird's counterexample [Baird et al., 1995]. Figures 3.2 and 3.3 show the MSBPE (averaged over 20 runs) as a function of the number of iterations. We can see that our gradient algorithms converge in these two counterexamples whereas TB(λ) and Retrace(λ) diverge.

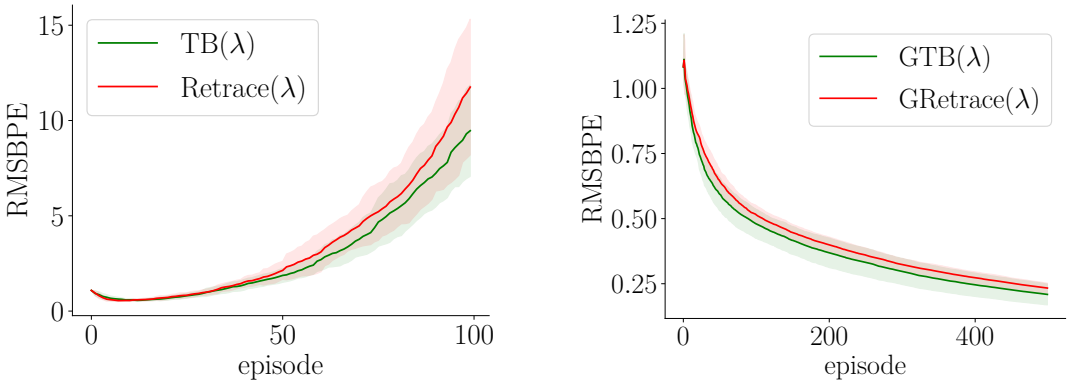

Figure 3.2: Baird's counterexample. The combination of linear function approximation with TB and Retrace leads to divergence (left panel) while the proposed gradient extensions GTB and GRetrace converge (right panel).

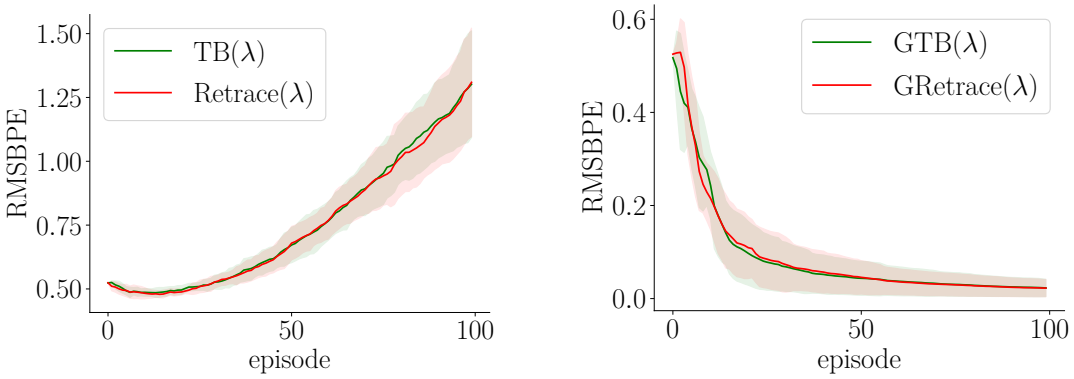

Figure 3.3: In the 2-states counterexample of section 3.4 showing that the gradient-based TB and RETRACE converge while TB and RETRACE diverge.

3.8.2 Comparison with existing methods

We also compared GTB(λ) and GRETRACE(λ) with two recent state-of-the-art convergent off-policy algorithms for action-value estimation and function approximation: GQ(λ) [Maei, 2011] and AB-TRACE(λ) [Mahmood et al., 2017]. As in Mahmood et al. [2017], we also consider a policy evaluation task in the Mountain Car domain. In order to better understand the variance inherent to each method, we designed the target policy and behavior policy in such a way that the importance sampling ratios can be as large as 30. We chose to describe state-action pairs by a 96-dimensional vector of features derived by tile coding [Sutton and Barto, 1998]. We ran each algorithm over all possible combinations of step-size values $(\alpha_k, \eta_k) \in [0.001, 0.005, 0.01, 0.05, 0.1]^2$ for 2000 episodes and reported their normalized mean squared errors (NMSE):

$$\textbf{NMSE}(\theta) = \frac{\|\Phi\theta - Q^\pi\|^2_\Xi}{\|Q^\pi\|^2_\Xi}$$

where Q^π is estimated by simulating the target policy and averaging the discounted cumulative rewards overs trajectories. As AB-TRACE(λ) and GRETRACE(λ) share both the same operator, we can evaluate them using the empirical $\textbf{MSPBE} = \frac{1}{2}||\hat{A}\theta + \hat{b}||^2_{\hat{M}^{-1}}$ where $\hat{A}$, $\hat{b}$ and $\hat{M}$ are Monte-Carlo estimates obtained by averaging $\hat{A}_k$, $\hat{b}_k$ and $\hat{M}_k$ defined in proposition 3 over 10000 episodes.

Figure 3.5 shows that the best empirical **MSPBE** achieved by AB-TRACE(λ) and GRETRACE(λ) are almost identical across value of λ. This result is consistent with the fact that they both minimize the **MSPBE** objective function. However, significant differences can be observed when computing the 5th percentiles of NMSE (over all possible combination of step-size values) for different values of

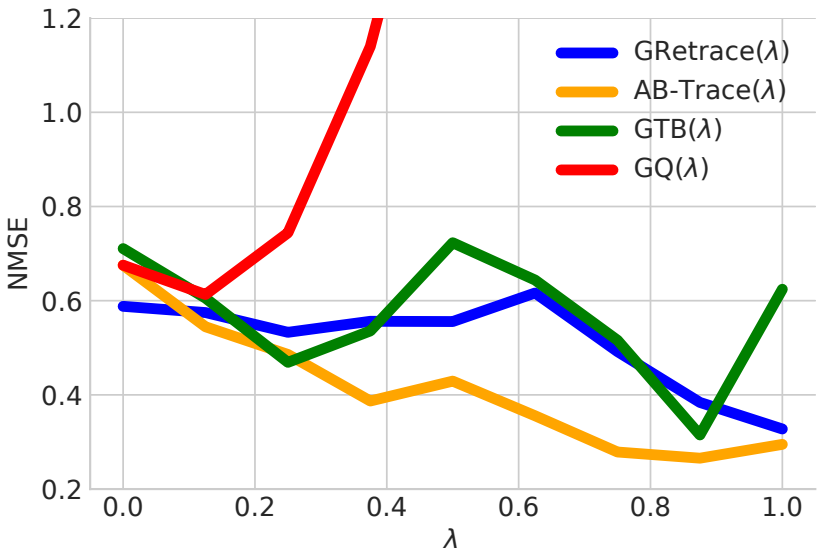

Figure 3.4: Each curves shows the 5th percentile of NMSE (over all possible combination of step-size values) achieved by each algorithm for different values of λ.

λ in Figure 3.4. When λ increases, the NMSE of GQ(λ) increases sharply due to increased influence of importance sampling ratios. This clearly demonstrate the variance issues of GQ(λ) in contrast with the other methods based on the TREE BACKUP and RETRACE returns (that are not using importance ratios). For intermediate values of λ, AB-TRACE(λ) performs better but its performance is matched by GRETRACE(λ) and TB(λ) for small and very large values of λ. In fact, AB-TRACE(λ) updates the function parameters θ as follows:

$$\theta_{k+1} = \theta_k - \alpha_k \left(\delta_k e_k - \Delta_k\right)$$

where $\Delta_k \triangleq \gamma w_k^\top e_k(\mathbb{E}_\pi \phi(s_{k+1}, .) - \lambda \sum_a \kappa(s_k, a)\mu(a \mid s_k)\phi(s_k, a))$ is a gradient correction term. When the instability is not an issue, the correction term could be very small and the update of θ would be essentially $\theta_{k+1} \sim \theta_k - \alpha_k \delta_k e_k$ so that θ_{k+1} follows the semi-gradient of the mean squared error $\|\Phi\theta - G_k^\lambda\|_\Xi^2$.
To better understand the errors of each algorithm and their robustness to step-size values, we propose the box plots shown in Figure 3.6. Each box plot shows the distribution of NMSE obtained by each algorithm for different values of λ. NMSE distributions are computed over all possible combinations of step-size values. GTB(λ) has the smallest variance as it scaled its return by the target probabilities which makes it conservative in its update even with large step-size values. GRETRACE(λ) tends to more more efficient than GTB(λ) since it could benefit from full returns. The latter observation agrees with the tabular case of TREE BACKUP and RETRACE [Munos et al., 2016]. Finally, we observe that AB-TRACE(λ) has lower error, but at the cost of increased variance with respect to

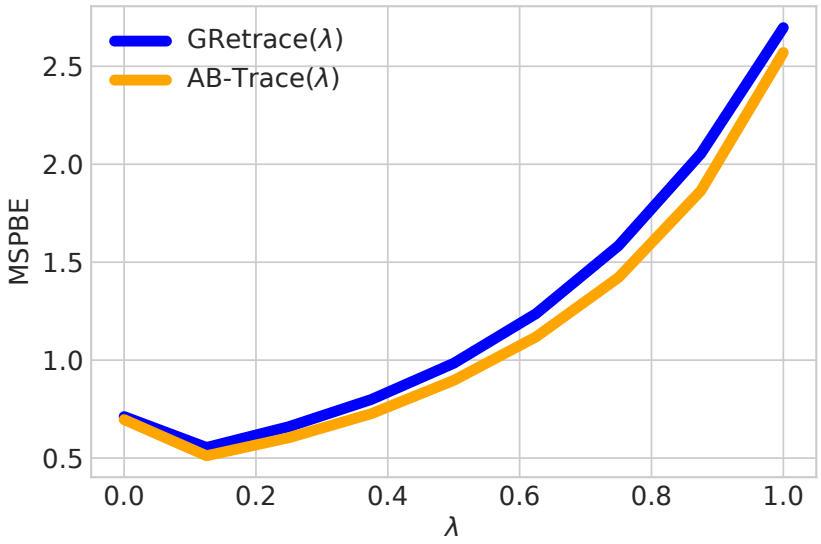

Figure 3.5: Comparison between the best empirical MSPBE obtained by each algorithm for different values of λ. Only GRETRACE(λ) and AB-TRACE(λ) are showed here because the other algorithms do not have the same operators and hence not the same **MSPBE**. Note that **MSPBE**s depend on λ. Thus, **MSPBE**s are not directly comparable across different values of λ. Both GRETRACE(λ) and AB-TRACE(λ) have very similar performances. AB-TRACE(λ) performs slightly better.

step-size values.

3.9 CONCLUSION

Our analysis highlighted for the first time the difficulties of combining the TREE BACKUP and RETRACE algorithms with function approximation. We addressed these issues by formulating gradient-based algorithm versions of these algorithms which minimize the mean-square projected Bellman error. Using a saddle-point formulation, we were also able to provide convergence guarantees and characterize the convergence rate of our algorithms GTB and GRETRACE. We also developed a novel analysis method which allowed us to establish a $O(1/k)$ convergence rate without having to use Polyak averaging or projections. Furthermore, our proof technique is general enough that we were able to apply it to the existing GTD and GTD2 algorithms. Our experiments finally suggest that the proposed GTB(λ) and GRETRACE (λ) are robust to step-size selection and have less variance than both GQ(λ) [Maei, 2011] and AB-TRACE(λ) [Mahmood et al., 2017].

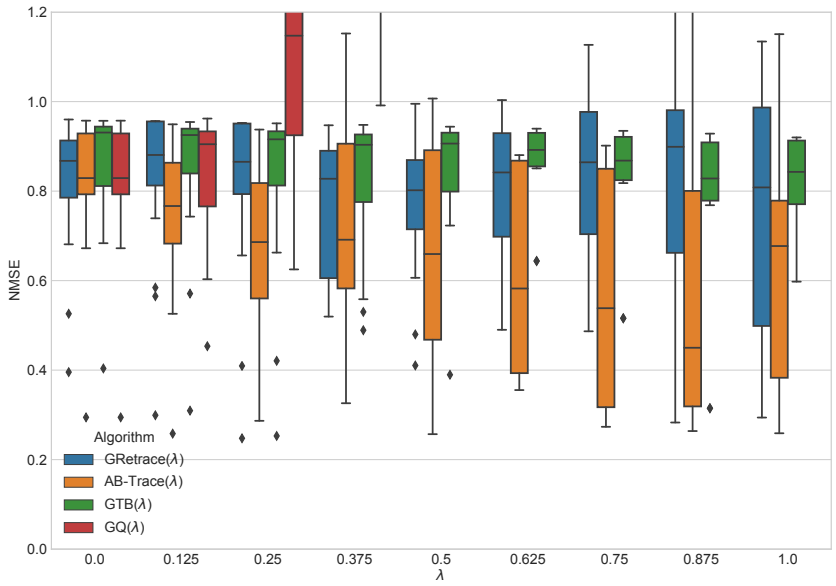

Figure 3.6: Comparison of empirical performance of GQ(λ), AB-TRACE(λ), GRETRACE(λ) and GTB(λ) on an off-policy evaluation task in Mountain Car domain. Each box plot shows the distribution of the NMSE achieved by each algorithm after 2000 episodes for different values of λ. NMSE distributions are computed over all the possible combinations of step-size values $(\alpha_k, \eta_k) \in [0.001, 0.005, 0.01, 0.05, 0.1]^2$.

4 Stable Policy Optimization via Off-Policy Divergence Regularization

4.1 Prologue to the Contribution

4.1.1 Article Details

This chapter is based on the article «*Stable Policy Optimization via Off-Policy Divergence Regularization*» [Touati et al., 2020c], jointly led with Amy Zhang and in collaboration with Joelle Pineau and Pascal Vincent. This paper was accepted as oral at UAI 2020. I share the first authorship with Amy Zhang. I was at the origin of the project's idea and the derivation of the new algorithm. I helped with the implementation of the algorithm and Amy performed the large-scale experiments of the paper.

4.1.2 Context

In the previous chapter, we saw how the discrepancy between the policy that generated the data to be learnt from (the behavior policy) and the policy being evaluated (the target policy) can lead to undesirable behavior. This behavior can be arbitrarily more complex in the control setting where we aim at learning a near-optimal policy. Some policy optimization algorithms control this distribution shift by ensuring that the improved policy stays in the vicinity of the current policy in term of their action probabilities.

In this work, we want to improve how these algorithms leverage off-policy data and stabilize the policy learning. Especially, we believe that the regularization based on immediate actions is not enough and that we should reason about long-term future state-action distribution. This is well supported when we look at formal analysis of offline RL algorithms such as approximate Value Iteration and Fitted Q-iteration [Munos, 2005, Munos and Szepesvári, 2008, Chen and Jiang, 2019, Touati and Vincent, 2020b]. These algorithms are concerned with finding a near-optimal policy given a fixed dataset of transitions and their analyses rely on strong assumptions about the distribution of batch data. Especially, they assume that the ratio between the induced state-action distribution of any non-stationary policy and the state-action distribution of the batch data is upper bounded by a constant, referred to as the *concentrability* coefficient. Unfortunately, this coefficient may be arbitrarily large or infinite, due the large policy class, which makes the error bounds of these analysis vacuous.

Even though our setting is easier than the fully batch setting as we allow ourselves to collect online data, we believe that, to use better off-policy data, we need to bias algorithmically the policy class in the improvement step to the one that is well-supported by the current off-policy data distribution.

4.1.3 Paper Abstract

Trust Region Policy Optimization (TRPO) and Proximal Policy Optimization (PPO) are among the most successful policy gradient approaches in deep reinforcement learning (RL). While these methods achieve state-of-the-art performance across a wide range of challenging tasks, there is room for improvement in the stabilization of the policy learning and how the off-policy data are used. In this paper we revisit the theoretical foundations of these algorithms and propose a new algorithm which stabilizes the policy improvement through a proximity term that constrains the discounted state-action visitation distribution induced by consecutive policies to be close to one another. This proximity term, expressed in terms of the divergence between the visitation distributions, is learned in an off-policy and adversarial manner. We empirically show that our proposed method can have a beneficial effect on stability and improve final performance in benchmark high-dimensional control tasks. Our code is available at https://github.com/facebookresearch/ppo-dice.

4.1.4 Recent Developments

In the fully batch setting, the work of [Liu et al., 2020] has the same underlying motivation as our work i.e constraining the learned policy not only in term of action distribution, like in previous batch RL methods, but rather in term of its coverage of the provided state-action distribution. They implement this idea as follows: given a batch of data, they learn a state-action density function, then, in both policy evaluation and improvement steps, they filter out state-action pairs whose estimated densities are below than some threshold. Theoretically, they show that they can find the approximately best policy with sufficient support in the batch data, without requiring a priori strong concentrability assumptions.

4.2 Introduction

Combined with deep neural networks as function approximators, policy gradient methods have enjoyed many empirical successes on RL problems such as video games [Mnih et al., 2016] and robotics [Levine et al., 2016]. Their recent success can be attributed to their ability to scale gracefully to high dimensional state-action spaces and complex dynamics.

The main idea behind policy gradient methods is to parametrize the policy and perform stochastic gradient ascent on the discounted cumulative reward directly [Sutton et al., 2000]. To estimate the gradient, we sample trajectories from the distribution induced by the policy. Due to the stochasticity of both policy and environment, variance of the gradient estimation can be very large, and lead to significant policy degradation.

Instead of directly optimizing the cumulative rewards, which can be challenging due to large variance, some approaches [Kakade and Langford, 2002, Azar et al., 2012, Pirotta et al., 2013, Schulman et al., 2015] propose to optimize a surrogate objective that can provide local improvements to the current policy at each iteration. The idea is that the advantage function of a policy π can produce a good estimate of the performance of another policy π' when the two policies give rise to similar state visitation distributions. Therefore, these approaches explicitly control the state visitation distribution shift between successive policies.

However, controlling the state visitation distribution shift requires measuring it, which is non-trivial. Direct methods are prohibitively expensive. Therefore, in order to make the optimization tractable, the aforementioned methods rely on constraining action probabilities by mixing policies [Kakade and Langford, 2002, Pirotta et al., 2013], introducing trust regions [Schulman et al., 2015, Achiam et al., 2017] or clipping the surrogate objective [Schulman et al., 2017, Wang et al., 2019b].

Our key motivation in this work is that constraining the probabilities of the immediate future actions might not be enough to ensure that the surrogate objective is still a valid estimate of the performance of the next policy and consequently might lead to instability and premature convergence. Instead, we argue that we should reason about the long-term effect of the policies on the distribution of the future states.

In particular, we directly consider the divergence between state-action visitation distributions induced by successive policies and use it as a regularization term added to the surrogate objective. This regularization term is itself optimized in an adversarial and off-policy manner by leveraging recent advances in off-policy policy evaluation [Nachum et al., 2019a] and off-policy imitation learning [Kostrikov et al., 2019]. We incorporate these ideas in the PPO algorithm in order to ensure safer policy learning and better reuse of off-policy data. We call our proposed method PPO-DICE.

The present paper is organized as follows: after reviewing conservative approaches for policy learning, we provide theoretical insights motivating our method. We explain how off-policy adversarial formulation can be derived to optimize the regularization term. We then present the algorithmic details of our proposed method. Finally, we show empirical evidences of the benefits of PPO-DICE as well as ablation studies.

4.3 Conservative Update Approaches

We consider a discounted MDP $(\mathcal{S}, \mathcal{A}, \gamma, P, r, \rho)$ with continuous state and action spaces $\mathcal{S}$ and $\mathcal{A}$. We assume in this chapter that all probability distributions are absolutely continuous with respect to the Lebesgue measure and we confound probability distributions with their probability density functions, by abuse of notation.

The goal of the agent is to find a policy π that maximizes the expected value from under the initial state distribution ρ:

$$\max_{\pi} J(\pi) \triangleq (1-\gamma)\mathbb{E}_{s\sim\rho}[V^{\pi}(s)].$$

We define the normalized discounted state visitation density d_{ρ}^{π} induced by a policy π:

$$d_{\rho}^{\pi}(s) \triangleq (1-\gamma)\sum_{t=0}^{\infty}\gamma^{t}\Pr(s_t = s \mid s_0 \sim \rho, \pi),$$

where $\Pr(s_t = s \mid s_0 \sim \rho, \pi)$ is the probability density that $s_t = s$, after we execute π for t steps, starting from initial state s_0 distributed according to ρ. Similarly, we define the discounted state-action visitation density $\mu_{\rho}^{\pi}(s,a)$ of policy π

$$\mu_{\rho}^{\pi}(s,a) \triangleq (1-\gamma)\sum_{t=0}^{\infty}\gamma^{t}\Pr(s_t = s, a_t = a \mid s_0 \sim \rho, \pi).$$

It is known [Puterman, 1990] that $\mu_{\rho}^{\pi}(s,a) = d_{\rho}^{\pi}(s)\cdot\pi(a \mid s)$ and that μ^{π} is characterized via: $\forall(s',a') \in \mathcal{S}\times\mathcal{A}$

$$\mu_{\rho}^{\pi}(s',a') = (1-\gamma)\rho(s')\pi(a' \mid s') + \gamma\int \pi(a' \mid s')P(s' \mid s,a)\mu_{\rho}^{\pi}(s,a)ds\,da,$$

Most policy training approaches in RL can be understood as updating a current policy π to a new improved policy π' based on the advantage function $A^{\pi} \triangleq Q^{\pi}(s,a) - V^{\pi}(s)$ or an estimate $\hat{A}$ of it. We review here some popular approaches that implement conservative updates in order to stabilize policy training.

First, let us state a key lemma from the seminal work of Kakade and Langford [2002] that relates the performance difference between two policies to the advantage function.

Lemma 1 (The performance difference lemma [Kakade and Langford, 2002])**.** *For all policies π and π',*

$$J(\pi') = J(\pi) + \mathbb{E}_{s\sim d_{\rho}^{\pi'}}\mathbb{E}_{a\sim\pi'(.|s)}\left[A^{\pi}(s,a)\right]. \tag{4.3.1}$$

This lemma implies that maximizing Equation (4.3.1) will yield a new policy π' with guaranteed performance improvement over a given policy π. Unfortunately, a naive direct application of this procedure would be prohibitively expensive since it requires estimating $d_\rho^{\pi'}$ for all π' candidates. To address this issue, Conservative Policy Iteration (CPI) [Kakade and Langford, 2002] optimizes a surrogate objective defined based on current policy π^i at each iteration i,

$$L_{\pi_i}(\pi') = J(\pi_i) + \mathbb{E}_{s\sim d_\rho^{\pi_i}} \mathbb{E}_{a\sim\pi'(.|s)} \left[A^{\pi_i}(s,a)\right], \tag{4.3.2}$$

by ignoring changes in state visitation distribution due to changes in the policy. Then, CPI returns the stochastic mixture $\pi_{i+1} = \alpha_i \pi_i^+ + (1-\alpha_i)\pi_i$ where $\pi_i^+ = \arg\max_{\pi'} L_{\pi_i}(\pi')$ is the greedy policy and $\alpha_i \in [0,1]$ is tuned to guarantee a monotonically increasing sequence of policies.

Inspired by CPI, the Trust Region Policy Optimization algorithm (TRPO) [Schulman et al., 2015] extends the policy improvement step to any general stochastic policy rather than just mixture policies. TRPO maximizes the same surrogate objective as CPI subject to a Kullback-Leibler (KL) divergence constraint that ensures the next policy π_{i+1} stays within δ-neighborhood of the current policy π_i:

$$\begin{aligned} \pi_{i+1} &= \arg\max_{\pi'} L_{\pi_i}(\pi') \\ \text{s.t} \quad & \mathbb{E}_{s\sim d_\rho^{\pi_i}} \left[D_{\mathrm{KL}}(\pi'(\cdot \mid s) \| \pi_i(\cdot \mid s))\right] \leq \delta, \end{aligned} \tag{4.3.3}$$

where D_{KL} is the Kullback–Leibler divergence. In practise, TRPO considers a differentiable parameterized policy $\{\pi_\theta, \theta \in \Theta\}$ and solves the constrained problem (4.3.3) in parameter space Θ. In particular, the step direction is estimated with conjugate gradients, which requires the computation of multiple Hessian-vector products. Therefore, this step can be computationally heavy.

To address this computational bottleneck, Proximal Policy Optimization (PPO) [Schulman et al., 2017] proposes replacing the KL divergence constrained objective (4.3.3) of TRPO by clipping the objective function directly as:

$$\begin{aligned} L_{\pi_i}^{\mathrm{clip}}(\pi') = \mathbb{E}_{(s,a)\sim\mu_\rho^{\pi_i}} \Big[& \min \left\{ A^{\pi_i}(s,a) \cdot \kappa_{\pi'/\pi_i}(s,a), \right. \\ & \left. A^{\pi_i}(s,a) \cdot \mathrm{clip}(\kappa_{\pi'/\pi_i}(s,a), 1-\epsilon, 1+\epsilon) \right\} \Big], \end{aligned} \tag{4.3.4}$$

where $\epsilon > 0$ and $\kappa_{\pi'/\pi_i}(s,a) = \frac{\pi'(s,a)}{\pi_i(s,a)}$ is the importance sampling ratio.

4.4 Theoretical Insights

In this section, we present the theoretical motivation of our proposed method.

At a high level, algorithms CPI, TRPO, and PPO follow similar policy update schemes. They optimize some surrogate performance objective ($L_{\pi_i}(\pi')$ for CPI and TRPO and $L_\pi^{\text{clip}}(\pi')$ for PPO) while ensuring that the new policy π_{i+1} stays in the vicinity of the current policy π_i. The vicinity requirement is implemented in different ways:

- CPI computes a sequence of stochastic policies that are mixtures between consecutive greedy policies.
- TRPO imposes a constraint on the KL divergence between old policy and new one ($\mathbb{E}_{s\sim d_\rho^{\pi_i}}\left[D_{\text{KL}}(\pi'(\cdot \mid s)\|\pi_i(\cdot \mid s))\right] \leq \delta$).
- PPO directly clips the objective function based on the value of the importance sampling ratio κ_{π'/π_i} between the old policy and new one.

Such conservative updates are critical for the stability of the policy optimization. In fact, the surrogate objective $L_{\pi_i}(\pi')$ (or its clipped version) is valid only in the neighbourhood of the current policy π_i, i.e, when π' and π_i visit all the states with similar probabilities. The following lemma more precisely formalizes this[1]:

Lemma 2. *For all policies π and π',*

$$\begin{aligned} J(\pi') &\geq L_\pi(\pi') - \epsilon^\pi D_{\text{TV}}(d_\rho^{\pi'}\|d_\rho^\pi) \\ &\geq L_\pi^{\text{clip}}(\pi') - \epsilon^\pi D_{\text{TV}}(d_\rho^{\pi'}\|d_\rho^\pi), \end{aligned} \tag{4.4.1}$$

where $\epsilon^\pi = \max_{s\in\mathcal{S}} \left|\mathbb{E}_{a\sim\pi'(\cdot|s)}\left[A^\pi(s,a)\right]\right|$ and D_{TV} is the total variation distance.

The proof is provided in appendix for completeness. Lemma 2 states that $L_\pi(\pi')$ (or $L_\pi^{\text{clip}}(\pi')$) is a sensible lower bound to $J(\pi')$ as long as π and π' are close in terms of total variation distance between their corresponding state visitation distributions $d_\rho^{\pi'}$ and d_ρ^π. However, the aforementioned approaches enforce closeness of π' and π in terms of their action probabilities rather than their state visitation distributions. This can be justified by the following inequality [Achiam et al., 2017]:

$$D_{\text{TV}}(d_\rho^{\pi'}\|d_\rho^\pi) \leq \frac{2\gamma}{1-\gamma}\mathbb{E}_{s\sim d_\rho^\pi}\left[D_{\text{TV}}(\pi'(.|s)\|\pi(.|s))\right]. \tag{4.4.2}$$

Plugging the last inequality (4.4.2) into (4.4.1) leads to the following lower bound:

$$J(\pi') \geq L_\pi(\pi') - \frac{2\gamma\epsilon^\pi}{1-\gamma}\mathbb{E}_{s\sim d_\rho^\pi}\left[D_{\text{TV}}(\pi'(.|s)\|\pi(.|s))\right]. \tag{4.4.3}$$

[1]The result is not novel, it can be found as intermediate step in proof of theorem 1 in Achiam et al. [2017], for example.

The obtained lower bound (4.4.3) is, however, clearly looser than the one in inequality (4.4.2). Lower bound (4.4.3) suffers from an additional multiplicative factor $\frac{1}{1-\gamma}$, which is the effective planning horizon. It is essentially due to the fact that we are characterizing a long-horizon quantity, such as the state visitation distribution $d_\rho^\pi(s)$, by a one-step quantity, such as the action probabilities $\pi(\cdot \mid s)$. Therefore, algorithms that rely solely on action probabilities to define closeness between policies should be expected to suffer from instability and premature convergence in long-horizon problems.

Furthermore, in the exact case if we take at iteration i, $\pi_{i+1} \leftarrow \arg\max_{\pi'} L_{\pi_i}(\pi') - \epsilon^{\pi_i} D_{\mathrm{TV}}(d_\rho^{\pi'} \| d_\rho^{\pi_i})$, then

$$\begin{aligned} J(\pi_{i+1}) &\geq L_{\pi_i}(\pi_{i+1}) - \epsilon^{\pi_i} D_{\mathrm{TV}}(d_\rho^{\pi_{i+1}} \| d_\rho^{\pi_i}) \\ &\geq L_{\pi_i}(\pi_i) && \text{(by optimality of } \pi_{i+1}) \\ &= J(\pi_i) \end{aligned}$$

Therefore, this provides a monotonic policy improvement, while TRPO suffers from a performance degradation that depends on the level of the trust region δ (see Proposition 1 in Achiam et al. [2017]).

It follows from our discussion that $D_{\mathrm{TV}}(d_\rho^{\pi'} \| d_\rho^{\pi})$ is a more natural proximity term to ensure safer and more stable policy updates. Previous approaches excluded using this term because we don't have access to $d_\rho^{\pi'}$, which would require executing π' in the environment. In the next section, we show how we can leverage recent advances in off-policy policy evaluation to address this issue.

4.5 Off-policy Formulation Of Divergences

In this section, we explain how divergences between state-visitation distributions can be approximated. This is done by leveraging ideas from recent works on off-policy learning [Nachum et al., 2019a, Kostrikov et al., 2019].

Consider two different policies π and π'. Suppose that we have access to state-action samples generated by executing the policy π in the environment, i.e, $(s, a) \sim \mu_\rho^\pi$. As motivated by the last section, we aim to estimate $D_{\mathrm{TV}}(d_\rho^{\pi'} \| d_\rho^{\pi})$ without requiring on-policy data from π'. Note that in order to avoid using importance sampling ratios, it is more convenient to estimate $D_{\mathrm{TV}}(\mu_\rho^{\pi'} \| \mu_\rho^{\pi})$, i.e, the total divergence between state-action visitation distributions rather than the divergence between state visitation distributions. This is still a reasonable choice

as $D_{\text{TV}}(d_\rho^{\pi'} \| d_\rho^{\pi})$ is upper bounded by $D_{\text{TV}}(\mu_\rho^{\pi'} \| \mu_\rho^{\pi})$ as shown below:

$$\begin{aligned} D_{\text{TV}}(d_\rho^{\pi'} \| d_\rho^{\pi}) &= \int_s \left| (d_\rho^{\pi'} - d_\rho^{\pi})(s) \right| ds \\ &= \int_s \left| \int_a (\mu_\rho^{\pi'} - \mu_\rho^{\pi})(s,a) da \right| ds \\ &\leq \int_s \int_a \left| (\mu_\rho^{\pi'} - \mu_\rho^{\pi})(s,a) \right| da\, ds \\ &= D_{\text{TV}}(\mu_\rho^{\pi'} \| \mu_\rho^{\pi}). \end{aligned}$$

The total variation distance belongs to a broad class of divergences known as ϕ-divergences [Sriperumbudur et al., 2009]. A ϕ-divergence is defined as,

$$D_\phi(\mu_\rho^{\pi'} \| \mu_\rho^{\pi}) = \mathbb{E}_{(s,a)\sim\mu_\rho^{\pi'}} \left[\phi \left(\frac{\mu_\rho^{\pi}(s,a)}{\mu_\rho^{\pi'}(s,a)} \right) \right], \tag{4.5.1}$$

where $\phi : [0, \infty) \to \mathbb{R}$ is a convex, lower-semicontinuous function and $\phi(1) = 0$. Well-known divergences can be obtained by appropriately choosing ϕ. These include the KL divergence ($\phi(t) = t \log(t)$), total variation distance ($\phi(t) = |t-1|$), χ^2-divergence ($\phi(t) = (t-1)^2$), etc. Working with the form of ϕ-divergence given in Equation (4.5.1) requires access to analytic expressions of both μ_ρ^{π} and μ_ρ^{π} as well as the ability to sample from $\mu_\rho^{\pi'}$. We have none of these in our problem of interest. To bypass these difficulties, we turn to the alternative variational representation of ϕ-divergences [Nguyen et al., 2009, Huang et al., 2017b] as

$$D_\phi(\mu_\rho^{\pi'} \| \mu_\rho^{\pi}) = \sup_{f:\mathcal{S}\times\mathcal{A}\to\mathbb{R}} \left[\mathbb{E}_{(s,a)\sim\mu_\rho^{\pi'}}[f(s,a)] - \mathbb{E}_{(s,a)\sim\mu_\rho^{\pi}}[\phi^\star \circ f(s,a)] \right], \tag{4.5.2}$$

where $\phi^\star(t) = \sup_{u\in\mathbb{R}}\{tu - \phi(u)\}$ is the convex conjugate of ϕ. The variational form in Equation (4.5.2) still requires sampling from $\mu_\rho^{\pi'}$, which we cannot do. To address this issue, we use a clever change of variable trick introduced by Nachum et al. [2019a]. Define $g : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as the fixed point of the following Bellman equation,

$$g(s,a) = f(s,a) + \gamma P^{\pi'} g(s,a), \tag{4.5.3}$$

where $\mathbb{P}^{\pi'}$ is the transition operator induced by π', defined as $\mathbb{P}^{\pi'} g(s,a) = \int \pi'(a' \mid s') \mathbb{P}(s' \mid s,a) g(s',a')$. g may be interpreted as the action-value function of the policy π' in a modified MDP which shares the same transition model $\mathbb{P}$ as the original MDP, but has f as the reward function instead of r. Applying the change of variable (4.5.3) to (4.5.2) and after some algebraic manipulation as

done in Nachum et al. [2019a], we obtain

$$D_\phi(\mu_\rho^{\pi'} \| \mu_\rho^\pi) = \sup_{g:\mathcal{S}\times\mathcal{A}\to\mathbb{R}} \Big[(1-\gamma)\mathbb{E}_{s\sim\rho, a\sim\pi'}[g(s,a)] - \mathbb{E}_{(s,a)\sim\mu_\rho^\pi}\Big[\phi^\star\Big((g-\gamma P^{\pi'} g)(s,a)\Big)\Big]\Big]. \tag{4.5.4}$$

Thanks to the change of variable, the first expectation over $\mu_\rho^{\pi'}$ in (4.5.2) is converted to an expectation over the initial distribution and the policy i.e $s \sim \rho(\cdot), a \sim \pi'(\cdot \mid s)$. Therefore, this new form of the ϕ-divergence in (4.5.4) is completely off-policy and can be estimated using only samples from the policy π.

Other possible divergence representations: Using the variational representation of ϕ-divergences was a key step in the derivation of Equation (4.5.4). But in fact any representation that admits a linear term with respect to $\mu_\rho^{\pi'}$ (i.e $\mathbb{E}_{(s,a)\sim\mu_\rho^{\pi'}}[f(s,a)]$) would work as well. For example, one can use the the Donkser-Varadhan representation [Donsker and Varadhan, 1983] to alternatively express the KL divergence as:

$$D_\phi(\mu_\rho^{\pi'} \| \mu_\rho^\pi) = \sup_{f:\mathcal{S}\times\mathcal{A}\to\mathbb{R}} \Big[\mathbb{E}_{(s,a)\sim\mu_\rho^{\pi'}}[f(s,a)] - \log\Big(\mathbb{E}_{(s,a)\sim\mu_\rho^{\pi}}[\exp(f(s,a))]\Big)\Big]. \tag{4.5.5}$$

The *log-expected-exp* in this equation makes the Donkser-Varadhan representation (4.5.5) more numerically stable than the variational one (4.5.4) when working with KL divergences. Because of its genericity for ϕ-divergences, we base the remainder of our exposition on (4.5.4). But it is straightforward to adapt the approach and algorithm to using (4.5.5) for better numerical stability when working with KL divergences specifically. Thus, in practice we will use the latter in our experiments with KL-based regularization, but not in the ones with χ^2-based regularization.

4.6 A Practical Algorithm Using Adversarial Divergence

We now turn these insights into a practical algorithm. The lower bounds in lemma 2, suggest using a regularized PPO objective[2] :

[2] Both regularized $L_{\pi_i}^{\text{clip}}$ and L_{π_i} are lower bounds on policy performance in Lemma 2. We

Algorithm 4 PPO-DICE

Initialisation: random initialize parameters θ_1 (policy), ψ_1 (discriminator) and ω_1 (value function).
for i=1, ... **do**
  Generate a batch of M rollouts $\{s_1^{(j)}, a_1^{(j)}, r_1^{(j)}, s_1^{(j)}, \ldots, s_T^{(j)}, a_T^{(j)}, r_T^{(j)}, s_{T+1}^{(j)}\}_{j=1}^{M}$ by executing policy π_{θ_i} in the environment for T steps.
  Estimate Advantage function: $\hat{A}(s_t^{(j)}, a_t^{(j)}) = \sum_{t=1}^{T} (\gamma\lambda)^{t-1}(r_t^{(j)} + \gamma V_{\omega_i}(s_{t+1}^{(j)}) - V_{\omega_i}(s_t^{(j)}))$
  Compute target value $y_t^{(j)} = r_t^{(j)} + \gamma r_{t+1}^{(j)} + \ldots + \gamma^{T+1-t} V_{\omega_i}(s_{T+1})$
  $\omega = \omega_i; \theta = \theta_i; \psi = \psi_i$
  for epoch n=1, ... N **do**
    for iteration k=1, ... K **do**
      // Compute discriminator loss:

$$\hat{L}_D(\psi, \theta) = \frac{1}{MT} \sum_{j=1}^{M} \sum_{t=1}^{T} \phi^\star \left(g_\psi(s_t^{(j)}, a_t^{(j)}) - \gamma g_\psi(s_{t+1}^{(j)}, a_{t+1}^{\prime(j)}) \right) - (1-\gamma) g_\psi(s_1^{(j)}, a_t^{\prime(j)})$$

      where $a_t^{\prime(j)} \sim \pi_\theta(\cdot \mid s_1^{(j)}), a_{t+1}^{\prime(j)} \sim \pi_\theta(\cdot \mid s_{t+1}^{(j)})$.
      // Update discriminator parameters: (using learning rate $c_\psi \eta$)
      $\psi \leftarrow \psi - c_\psi \eta \nabla_\psi \hat{L}_D(\psi, \theta)$;
    end for
    // Compute value loss:
    $\hat{L}_V(\omega) = \frac{1}{MT} \sum_{j=1}^{M} \sum_{t=1}^{T} \left(V_\omega(s_t^{(j)}) - y_t^{(j)} \right)^2$
    // Compute PPO clipped loss:

$$\hat{L}^{\text{clip}}(\theta) = \frac{1}{MT} \sum_{j=1}^{M} \sum_{t=1}^{T} \min \{ \hat{A}(s_t^{(j)}, a_t^{(j)}) \kappa_{\pi_\theta / \pi_{\theta_i}}(s_t^{(j)}, a_t^{(j)}), \hat{A}(s_t^{(j)}, a_t^{(j)}) \cdot \text{clip}(\kappa_{\pi_\theta / \pi_{\theta_i}}(s_t^{(j)}, a_t^{(j)}), 1-\epsilon, 1+\epsilon) \}$$

    // Update parameters: (using learning rate η)
    $\omega \leftarrow \omega - \eta \nabla_\omega \hat{L}_V(\omega)$;
    $\theta \leftarrow \theta + \eta \nabla_\theta (\hat{L}^{\text{clip}}(\theta) + \lambda \cdot \hat{L}_D(\psi, \theta))$ (if reparametrization trick applicable, else gradient step on Eq. (4.6.7))
  end for
  $\omega_{i+1} = \omega; \theta_{i+1} = \theta; \psi_{i+1} = \psi$
end for

use $L_{\pi_i}^{\text{clip}}$ rather than L_{π_i} because we expect it to work better as the clipping already provides some constraint on action probabilities. Also this will allow a more direct empirical assessment of what the regularization brings compared to vanilla PPO.

$L_\pi^{\text{clip}}(\pi') - \lambda D_{\text{TV}}(d_\rho^{\pi'} \| d_\rho^{\pi})$, where λ is a regularization coefficient. If in place of the total variation we use the off-policy formulation of ϕ-divergence $D_\phi(\mu_\rho^{\pi'} \| \mu_\rho^{\pi})$ as detailed in Equation (4.5.4), our main optimization objective can be expressed as the following min-max problem:

$$\max_{\pi'} \min_{g:\mathcal{S}\times\mathcal{A}\to\mathbb{R}} L_{\pi_i}^{\text{clip}}(\pi') - \lambda\Big((1-\gamma)\mathbb{E}_{s\sim\rho,a\sim\pi'}[g(s,a)] - \mathbb{E}_{(s,a)\sim\mu_\rho^{\pi_i}}\left[\phi^\star\left((g-\gamma P^{\pi'} g)(s,a)\right)\right]\Big), \tag{4.6.1}$$

When the inner minimization over g is fully optimized, it is straightforward to show – using the score function estimator – that the gradient of this objective with respect to π is (proof is provided in appendix):

$$\nabla_{\pi'} L_{\pi_i}^{\text{clip}}(\pi') - \lambda\Big((1-\gamma)\mathbb{E}_{\substack{s\sim\rho\\a\sim\pi'}}[g(s,a)\nabla_{\pi'}\log\pi'(a\mid s)] \tag{4.6.2}$$

$$+\gamma\mathbb{E}_{(s,a)\sim\mu_\rho^{\pi_i}}\left[\frac{\partial\phi^\star}{\partial t}\left((g-\gamma P^{\pi'}g)(s,a)\right)\mathbb{E}_{\substack{s'\sim P(\cdot|s,a),\\a'\sim\pi'(\cdot|s')}}\left[g(s',a')\nabla_{\pi'}\log\pi'(a'\mid s')\right]\right]\Big).$$

Furthermore, we can use the reparametrization trick if the policy π is parametrized by a Gaussian, which is usually the case in continuous control tasks. We call the resulting new algorithm PPO-DICE, (detailed in Algorithm 4), as it uses the clipped loss of PPO and leverages the DIstribution Correction Estimation idea from Nachum et al. [2019a].

In the min-max objective (4.6.1), g plays the role of a discriminator, as in Generative Adversarial Networks (GAN) [Goodfellow et al., 2014]. The policy π' plays the role of a generator, and it should balance between increasing the likelihood of actions with large advantage versus inducing a state-action distribution that is close to the one of π_i.

As shown in Algorithm 4, both policy and discriminator are parametrized by neural networks π_θ and g_ψ respectively. We estimate the objective (4.6.1) with samples from $\pi_i = \pi_{\theta_i}$ as follows. At a given iteration i, we generate a batch of M rollouts $\{s_1^{(j)}, a_1^{(j)}, r_1^{(j)}, s_1^{(j)}, \ldots, s_T^{(j)}, a_T^{(j)}, r_T^{(j)}, s_{T+1}^{(j)}\}_{j=1}^M$ by executing the policy π_i in the environment for T steps. Similarly to the PPO procedure, we learn a value function V_ω by updating its parameters ω with gradient descent steps, optimizing the following squared error loss:

$$\hat{L}_V(\omega) = \frac{1}{MT}\sum_{j=1}^{M}\sum_{t=1}^{T}\left(V_\omega(s_t^{(j)}) - y_t^{(j)}\right)^2, \tag{4.6.3}$$

where $y_t^{(j)} = r_t^{(j)} + \gamma r_{t+1}^{(j)} + \ldots + \gamma^{T+1-t} V_\omega(s_{T+1})$. Then, to estimate the advantage, we use the truncated generalized advantage estimate

$$\hat{A}(s_t^{(j)}, a_t^{(j)}) = \sum_{t=1}^{T}(\gamma\lambda)^{t-1}(r_t^{(j)} + \gamma V_\omega(s_{t+1}^{(j)}) - V_\omega(s_t^{(j)})). \tag{4.6.4}$$

This advantage estimate is used to compute an estimate of $L^{\text{clip}}_{\pi_i}$ given by:

$$\hat{L}^{\text{clip}}(\theta) = \frac{1}{MT}\sum_{j=1}^{M}\sum_{t=1}^{T}\min\Big\{\hat{A}(s_t^{(j)}, a_t^{(j)})\kappa_{\pi_\theta/\pi_i}(s_t^{(j)}, a_t^{(j)}), \tag{4.6.5}$$
$$\hat{A}(s_t^{(j)}, a_t^{(j)}) \cdot \text{clip}(\kappa_{\pi_\theta/\pi_i}(s_t^{(j)}, a_t^{(j)}), 1-\epsilon, 1+\epsilon)\Big\}$$

The parameters ψ of the discriminator are learned by gradient descent on the following empirical version of the regularization term in the min-max objective (4.6.1)

$$\hat{L}_D(\psi, \theta) = \frac{-1}{MT}\sum_{j=1}^{M}\sum_{t=1}^{T}(1-\gamma)g_\psi(s_1^{(j)}, a_t'^{(j)}) - \phi^\star\left(g_\psi(s_t^{(j)}, a_t^{(j)}) - \gamma g_\psi(s_{t+1}^{(j)}, a_{t+1}'^{(j)})\right), \tag{4.6.6}$$

where $a_t'^{(j)} \sim \pi_\theta(\cdot \mid s_1^{(j)})$ and $a_{t+1}'^{(j)} \sim \pi_\theta(\cdot \mid s_{t+1}^{(j)})$.

If the reparametrization trick is applicable (which is almost always the case for continuous control tasks), the parameters θ of the policy are updated via gradient ascent on the objective $\hat{L}^{\text{clip}}(\theta) + \lambda\hat{L}_D(\psi, \theta)$ as we can backpropagate gradient though the action sampling while computing $\hat{L}_D(\psi, \theta)$ in Equation (4.6.6). Otherwise, θ are updated via gradient ascent on the following objective:

$$\hat{L}^{\text{clip}}(\theta) - \frac{\lambda}{MT}\sum_{j=1}^{M}\sum_{t=1}^{T}(1-\gamma)g_\psi(s_1^{(j)}, a_t'^{(j)})\log\pi_\theta(a_t'^{(j)} \mid s_1^{(j)}) \tag{4.6.7}$$
$$+ \gamma\frac{\partial\phi^\star}{\partial t}\left(g_\psi(s_t^{(j)}, a_t^{(j)}) - \gamma g_\psi(s_{t+1}^{(j)}, a_{t+1}'^{(j)})\right) \cdot g_\psi(s_{t+1}^{(j)}, a_{t+1}'^{(j)})\log\pi_\theta(a_{t+1}'^{(j)} \mid s_{t+1}^{(j)})$$

Note that the gradient of this equation with respect to θ corresponds to an empirical estimate of the score function estimator we provided in Equation 4.6.2.

We train the value function, policy, and discriminator for N epochs using M rollouts of the policy π_i. We can either alternate between updating the policy and the discriminator, or update g_ψ for a few steps M before updating the policy. We found that the latter worked better in practice, likely due to the fact that the target distribution $\mu_\rho^{\pi_i}$ changes with every iteration i. We also found that increasing the learning rate of the discriminator by a multiplicative factor c_ψ of the learning rate for the policy and value function η improved performance.

Choice of divergence: The algorithmic approach we just described is valid with any choice of ϕ-divergence for measuring the discrepancy between state-visitation distributions. It remains to choose an appropriate one. While Lemma 2 advocates the use of total variation distance ($\phi(t) = |t-1|$), it is notoriously hard to train high dimensional distributions using this divergence (see Kodali

et al. [2017] for example). Moreover, the convex conjugate of $\phi(t) = |t-1|$ is $\phi^\star(t) = t$ if $|t| \leq \frac{1}{2}$ and $\phi^\star(t) = \infty$ otherwise. This would imply the need to introduce an extra constraint $\|g - P^\pi g\|_\infty \leq \frac{1}{2}$ in the formulation (4.5.4), which may be hard to optimize.

Therefore, we will instead use the KL divergence ($\phi(t) = t\log(t), \phi^\star(t) = \exp(t-1)$). This is still a well justified choice as we know that $D_{\text{TV}}(\mu_\rho^{\pi'} \| \mu_\rho^\pi) \leq \sqrt{\frac{1}{2} D_{\text{KL}}(\mu_\rho^{\pi'} \| \mu_\rho^\pi)}$ thanks to Pinsker's inequality. We will also try χ^2-divergence ($\phi(t) = (t-1)^2$) that yields a squared regularization term.

4.7 Related Work

Constraining policy updates, in order to minimize the information loss due to policy improvement, has been an active area of investigation. Kakade and Langford [2002] originally introduce CPI by maximizing a lower bound on the policy improvement and relaxing the greedification step through a mixture of successive policies. Pirotta et al. [2013] build on Kakade and Langford [2002] refine the lower bounds and introduce a new mixture scheme. Moreover, CPI inspired some popular Deep RL algorithms such as TRPO [Schulman et al., 2015] and PPO [Schulman et al., 2015], Deep CPI [Vieillard et al., 2019] and MPO [Abdolmaleki et al., 2018]. The latter uses similar updates to TRPO/PPO in the parametric version of its E-step. So, our method can be incorporated to it.

Our work is related to regularized MDP literature [Neu et al., 2017, Geist et al., 2019]. Shannon Entropic regularization is used in value iteration scheme [Haarnoja et al., 2017, Dai et al., 2018] and in policy iteration schemes [Haarnoja et al., 2018]. Note that all the mentioned works employ regularization on the action probabilities. Recently, Wang et al. [2019a] introduce divergence-augmented policy optimization where they penalize the policy update by a Bregman divergence on the state visitation distributions, motivated the mirror descent method. While their framework seems general, it doesn't include the divergences we employ in our algorithm. In fact, their method enables the use of the *conditional* KL divergence between state-action visitations distribution defined by $\int \mu_\rho^\pi(s,a) \log \frac{\pi(a|s)}{\pi'(a|a)}$ and not the KL divergence $\int \mu_\rho^\pi(s,a) \log \frac{\mu_\rho^\pi(s,a)}{\mu_\rho^{\pi'}(s,a)}$. Note the action probabilities ratio inside the log in the conditional KL divergence allows them to use the policy gradient theorem, a key ingredient in their framework, which cannot be done for the KL divergence.

Our work builds on recent off-policy approaches: DualDICE [Nachum et al., 2019a] for policy evaluation and ValueDICE [Kostrikov et al., 2019] for imitation learning. Both use the off-policy formulation of KL divergence. The former uses the formulation to estimate the ratio of the state visitation distributions

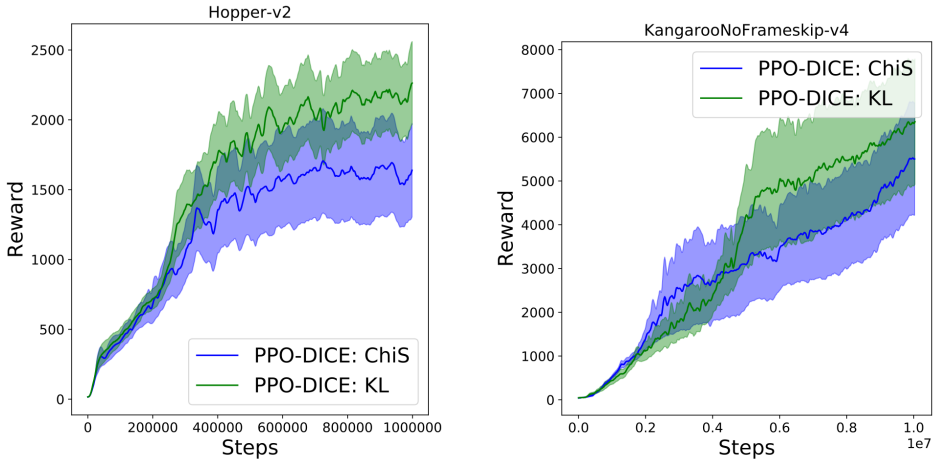

Figure 4.1: Comparison of χ^2 and KL divergences for PPO-DICE for two randomly selected environments in OpenAI Gym MuJoCo and Atari, respectively. We see that KL performs better than χ^2 in both settings. Performance plotted across 10 seeds with 1 standard error shaded.

under the target and behavior policies. Whereas, the latter learns a policy by minimizing the divergence.

The closest related work is the recently proposed AlgaeDICE [Nachum et al., 2019b] for off-policy policy optimization. They use the divergence between state-action visitation distribution induced by π and a behavior distribution, motivated by similar techniques in Nachum et al. [2019a]. However, they incorporate the regularization to the dual form of policy performance $J(\pi) = \mathbb{E}_{(s,a)\sim\mu_\rho^\pi}[r(s,a)]$ whereas we consider a surrogate objective (lower bound on the policy performance). Moreover, our method is online off-policy in that we collect data with each policy found in the optimization procedure, but also use previous data to improve stability. Whereas, their algorithm is designed to learn a policy from a fixed dataset collected by behaviour policies. Further comparison with AlgaeDICE is provided in appendix.

4.8 Experiments And Results

We use the PPO implementation by Kostrikov [2018] as a baseline and modify it to implement our proposed PPO-DICE algorithm. We run experiments on a randomly selected subset of environments in the Atari suite [Bellemare et al., 2013a] for high-dimensional observations and discrete action spaces, as well as on the OpenAI Gym [Brockman et al., 2016] MuJoCo environments, which have continuous state-action spaces. All shared hyperparameters are set at the same

values for both methods, and we use the hyperparameter values recommended by Kostrikov [2018] for each set of environments, Atari and MuJoCo [3].

4.8.1 Important Aspects Of PPO-DICE

Choice of Divergence

We conducted an initial set of experiments to compare two different choices of divergences, KL and χ^2, for the regularization term of PPO-DICE. Figure 4.1 shows training curves for one continuous action and one discrete action environment. There, as in the other environments in which we run this comparison, KL consistently performed better than χ^2. We thus opted to use KL divergence in all subsequent experiments.

Effect of Varying λ

Next we wanted to evaluate the sensitivity of our method to the λ parameter that controls the strength of the regularization. We examine in Figure 4.2 the performance of PPO-DICE when varying λ. There is a fairly narrow band for `Hopper-v2` that performs well, between 0.01 and 1. Theory indicates that the proper value for λ is the maximum of the absolute value of the advantages (see Lemma 2). This prompted us to implement an adaptive approach, where we compute the 90th percentile of advantages within the batch (for stability), which we found performed well across environments. To avoid introducing an additional hyperparameter by tuning λ, we use the adaptive method for subsequent experiments.

[3] Code: https://github.com/facebookresearch/ppo-dice

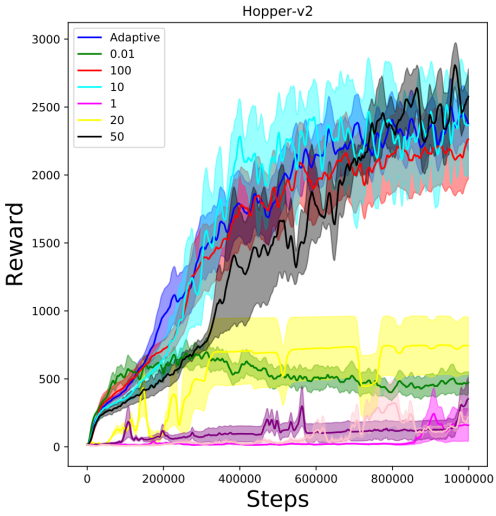

Figure 4.2: Varying λ in `Hopper_v2`, 10 seeds, 1 standard error shaded. PPO-DICE is somewhat sensitive to λ value, but the theoretically-motivated adaptive version works well.

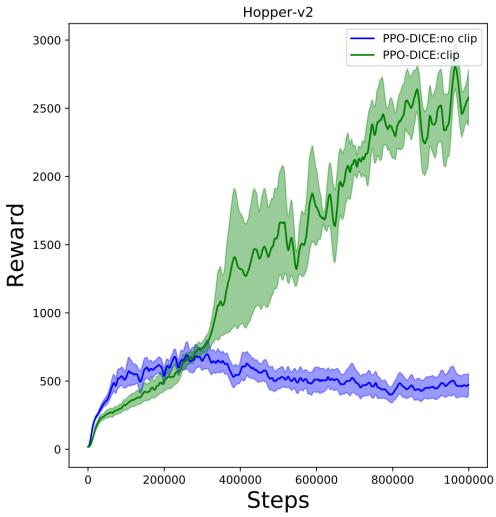

Figure 4.3: Comparison of PPO-DICE with clipped loss L^{clip} and without L. We see that clipping the action loss is crucial for good performance.

Game	PPO	PPO-DICE
AirRaid	4305.0 ± 638.15	**5217.5 ± 769.19**
Asterix	4300.0 ± 169.31	**6200.0 ± 754.10**
Asteroids	1511.0 ± 125.03	**1653.0 ± 112.20**
Atlantis	2120400.0 ± 471609.93	**3447433.33 ± 100105.82**
BankHeist	1247.0 ± 21.36	**1273.33 ± 7.89**
BattleZone	**29000.0 ± 2620.43**	19000.0 ± 2463.06
Carnival	3243.33 ± 369.51	3080.0 ± 189.81
ChopperCommand	566.67 ± 14.91	**900.0 ± 77.46**
DoubleDunk	-6.0 ± 1.62	**-4.0 ± 1.26**
Enduro	1129.9 ± 73.18	**1308.33 ± 120.09**
Freeway	32.33 ± 0.15	32.0 ± 0.00
Frostbite	**639.0 ± 334.28**	296.67 ± 5.96
Gopher	1388.0 ± 387.65	1414.0 ± 417.84
Kangaroo	4060.0 ± 539.30	**6650.0 ± 1558.16**
Phoenix	**12614.0 ± 621.71**	11676.67 ± 588.24
Robotank	7.8 ± 1.33	**12.1 ± 2.91**
Seaquest	1198.0 ± 128.82	1300.0 ± 123.97
TimePilot	5070.0 ± 580.53	**7000.0 ± 562.32**
Zaxxon	**7110.0 ± 841.60**	6130.0 ± 1112.48

Table 4.1: Mean final reward and 1 standard error intervals across 10 seeds for Atari games evaluated at 10M steps.

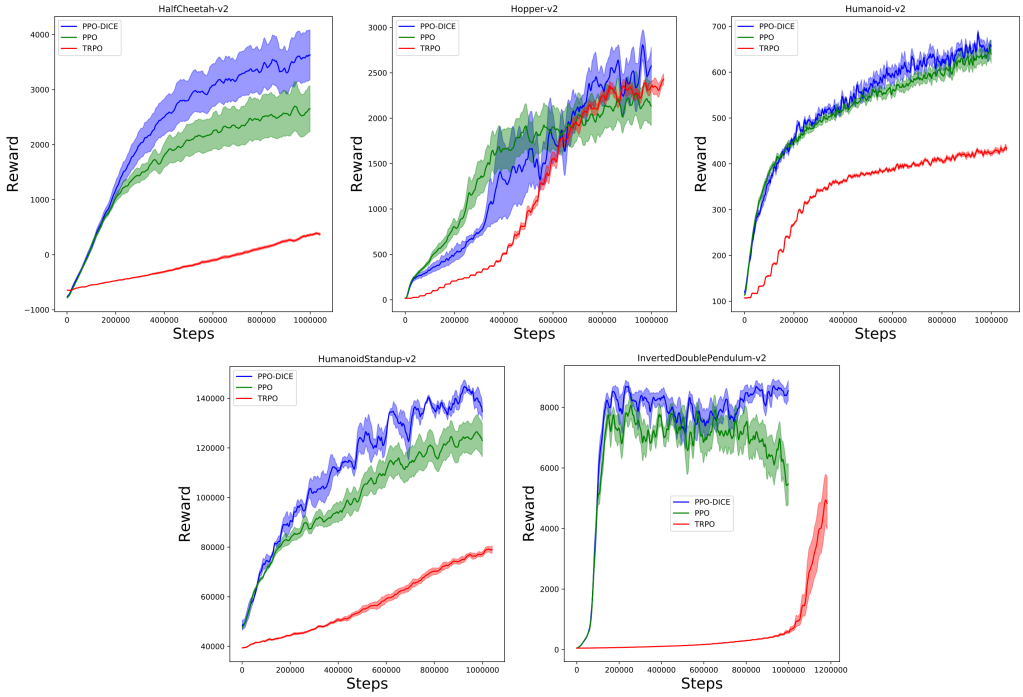

Figure 4.4: Results from OpenAI Gym MuJoCo suite in more complex domains, with 10 seeds and 1 standard error shaded. Results on the full suite of environments can be found in B.4.

Importance of Clipping the Action Loss

We earlier mentioned (see footnote 2) two possible forms of our regularized objective: one with clipped action loss L^{clip} and one without L. Clipping the action loss was an extra regularizing measure proposed in PPO [Schulman et al., 2017]. For our algorithm also, we hypothesized that it would be important for providing additional constraints on the policy update to stay within the trust region. Figure 4.3 confirms this empirically: we see the effect on our method of clipping the action loss versus keeping it unclipped. Initially, not having the additional regularization allows it to learn faster, but it soon crashes, showing the need for clipping to reduce variance in the policy update.

4.8.2 Results On Atari

Given our above observations we settled on using a KL-regularized L^{clip}, with the adaptive method for λ that we explained Section 4.8.1. We run PPO-DICE on randomly selected environments from Atari. We tuned two additional hyperparameters, the learning rate for the discriminator and the number of discriminator optimization steps per policy optimization step. We found that $K = 5$ discriminator optimization steps per policy optimization step performed well.

Fewer steps showed worse performance because the discriminator was not updating quickly enough, while more optimization steps introduced instability from the discriminator overfitting to the current batch. We also found that increasing the discriminator learning rate to be $c_\psi = 10\times$ the policy learning rate helped most environments. We used the same hyperparameters across all environments. Results are shown in Figure 4.1. We see that PPO-DICE significantly outperforms PPO on a majority of Atari environments.

4.8.3 Results On OpenAI Gym MuJoCo

For the OpenAI Gym MuJoCo suite, we also used $K = 5$ discriminator optimization steps per policy optimization step, and $c_\psi = 10\times$ learning rate for the discriminator in all environments. We selected 5 of the more difficult environments to showcase in this present (4.4), but additional results on the full suite and all hyperparameters used can be found in B.4. We again see improvement in performance in the majority of environments with PPO-DICE compared to PPO and TRPO.

4.9 Conclusion

In this work, we have argued that using the action probabilities to constrain the policy update is a suboptimal approximation to controlling the state visitation distribution shift. We then demonstrate that using the recently proposed DIstribution Correction Estimation idea [Nachum et al., 2019a], we can directly compute the divergence between the state-action visitation distributions of successive policies and use that to regularize the policy optimization objective instead. Through carefully designed experiments, we have shown that our method beats PPO in most environments in Atari [Bellemare et al., 2013a] and OpenAI Gym MuJoCo [Brockman et al., 2016] benchmarks.

5 Online Learning in Smooth Markov Decision Processes

5.1 Protologue To The Contribution

5.1.1 Article Details

This paper is based on the preprint *Zooming for Efficient Model-Free Reinforcement Learning in Metric Spaces* [Touati et al., 2020b]. This is a joint work with Adrien Ali Taiga and Marc G. Bellemare. I am the main author. I performed the derivation of the new algorithm and its theoretical analysis.

5.1.2 Context

In previous chapters, we were concerned with better exploiting the off-policy data but we didn't talk that much about the data collection itself. While recent RL empirical successes have been demonstrated in simulated environments where data can be generated indefinitely up to computational limits, bringing RL to real systems require us to focus on statistical efficiency as real data collection is costly and limited by the physical context. A statistically efficient agent must carefully balance between exploiting its current knowledge and exploring towards informative data.

My interest in this line of research was triggered by an inspiring talk given by Benjamin Van Roy at Mcgill university about what he coined as *deep exploration*. Deep exploration means exploration which is directed over multiple time steps and takes into account the consequences of an action on future learning. In Osband et al. [2019], his group developed a new approach consisting in choosing actions that are greedy with respect a randomly drawn value function from a proxy of the posterior distribution over value functions. As a first contribution on the topic of exploration, I proposed in Touati et al. [2020a] to scale the idea of randomized value functions to deep RL by leveraging recent advances in variational inference to account for the uncertainty of estimates

Now, the missing piece is that even though randomized value functions offer a promising computationally tractable approach for high dimensional state spaces, the algorithm comes with no theoretical guarantee when combined with function approximation. Especially, it is only shown to be provably efficient with tabular representation. In fact, there was (and it is still true at some extent) a huge gap between the RL theory and practise: despite the wealth of research

into provably efficient reinforcement learning algorithms, most works focus on tabular representation and thus struggle to handle exponentially or infinitely large state-action spaces. This present work is a step towards bridging this gap.

To make exploration with large state-action space intractable, we need to make some structural assumption. In this work, we assume that the MDP is smooth: a natural metric characterizes the proximity between different states and actions and the optimal action-value function is Lipschitz with respect to this metric. This is also known as the metric space assumption.

5.1.3 Paper Abstract

Despite the wealth of research into provably efficient reinforcement learning algorithms, most works focus on tabular representation and thus struggle to handle exponentially or infinitely large state-action spaces. In this paper, we consider episodic reinforcement learning with a metric state-action space. We propose ZOOMRL, an online model-free algorithm that learns an adaptive discretization of the joint space by *zooming* in more promising and frequently visited regions while carefully balancing the exploitation-exploration trade-off. We show that ZOOMRL achieves a worst-case regret $\tilde{O}(H^{\frac{5}{2}}K^{\frac{d+1}{d+2}})$ where H is the planning horizon, K is the number of episodes and d is the covering dimension of the space with respect to the metric. Moreover, our algorithm enjoys improved metric-dependent guarantees that reflect the geometry of the underlying space. Finally, we show that our algorithm is robust to small misspecification errors.

5.1.4 Recent Developments

The bounds presented in this work are instance-independent guarantees. A recent work Cao and Krishnamurthy [2020] shows that adaptive partitioning approaches achieve problem-dependent guarantees which depend on the *zooming dimension* instead of the covering dimension of the space. The zooming dimension is a measure of near-optimal regions. In benign instances where the near-optimal regions concentrate to a low dimensional manifold, the zooming dimension could be much smaller than the covering dimension, which enables sharper regret bounds.

Model-based algorithms for metric spaces was proposed in Domingues et al. [2020b] using non-parametric kernel estimators and in Sinclair et al. [2020] using adaptive discretization but their regret bounds have sub-optimal dependence on K.

5.2 INTRODUCTION

In the regime of MDPs with a finite state-action space, the OFU principle has been successfully implemented and efficient algorithms typically achieve regret that scales sublinearly with the number of discrete states and the number of discrete actions. This precludes applying them to arbitrarily large state-action spaces. On the other hand, MDPs with continuous state-action spaces have been an active area of investigation [Ortner and Ryabko, 2012, Lakshmanan et al., 2015, Song and Sun, 2019]. A common theme is to assume some structure knowledge, such as the existence of similarity metric between state-action pairs, and then to use a uniform discretisation of the space or nearest-neighbor approximators.

In this work, we focus on the finite-horizon MDP formalism with an unknown transition kernel. We suppose that the state-action space is equipped by a metric that characterizes the proximity between different states and actions. Such metrics have been studied in previous work for state aggregation [Ferns et al., 2004, Ortner, 2007]. We assume that the optimal action-value function is Lipschitz continuous with respect to this metric, which means that state-action pairs that are close to each other have similar optimal values.

We propose an online model-free RL algorithm, ZOOMRL, that actively explores the state-action space by learning on-the-fly an adaptive partitioning. Algorithms based on uniform partitions, such as the works in Ortner and Ryabko [2012] and Song and Sun [2019], disregard the shape of the optimal value function and thus could waste effort in partitioning irrelevant regions of the space. Moreover, the granularity of the partition should be tuned and it depends on the time horizon and the covering dimension of joint space. In contrast, ZOOMRL is able to take advantage of the structure of the problem's instance at hand by adjusting the discretisation to frequently visited and high-rewarding regions to get better estimates. Zooming approaches have been successfully applied in Lipschitz bandits [Kleinberg et al., 2008] and continuous contextual bandits [Slivkins, 2014]. However, in the bandit setting, an algorithm's cumulative regret can be easily decomposed into regret incurred in each sub-partition which is controlled by the size of the sub-partition itself. In contrast, in the reinforcement learning setting, the errors are propagated through iterations and we need to carefully control how they accumulate over iterations and navigate through sub-partitions. We show that ZOOMRL achieves a worst-case regret $\tilde{O}(H^{\frac{5}{2}}K^{\frac{d+1}{d+2}})$ where H is the planning horizon, K is the number of episodes and d is the covering dimension of the space with respect to the metric. Moreover, ZOOMRL enjoys an improved metric-dependent guarantee that reflects the geometry of the underlying space and whose scaling in terms of K is optimal as it matches the lower bound in continuous contextual bandit [Slivkins, 2014] when $H = 1$. Finally, we study how our algorithm cope with the misspecified setting (Assump-

tion 5). We show that it is robust to small misspecification error as it suffers only from an additional regret term $O(HK\epsilon)$ if the true optimal action-value function is Lipschitz up to an additive error uniformly bounded in absolute value by ϵ.

5.3 Related Work

Exploration in metric spaces: There have been several recent works that study exploration in continuous state-action MDPs under different structured assumptions. Kakade et al. [2003] assume a local continuity of the reward function and the transition kernel with respect to a given metric. They propose a generalization of the E^3 algorithm of Kearns and Singh [2002] to metric spaces. Their sample complexity depends on the covering number of the space under the continuity metric instead of the number of the states. However, their algorithm requires access to an approximate planning oracle. Lattimore et al. [2013] assume that the true transition kernel belongs to a finite or compact hypothesis class. Their algorithm consists in maintaining a set of transitions models and pruning it over time by eliminating the provable implausible models. They establish a sample complexity that depends polynomially on the cardinality or covering number of the model class. Pazis and Parr [2013] consider a continuous state-action MDP, develop a nearest-neighbor based algorithm under the assumption that all Q-functions encountered are Lipschitz continuous, showing a sample complexity that depends on an approximate covering number. Ortner and Ryabko [2012] develop a model-based algorithm that combines state aggregation with the standard UCRL2 algorithm [Jaksch et al., 2010] under the assumption of Lipschitz or Hölder continuity of rewards and transition kernel and they establish a regret bound scaling in $K^{\frac{2d+1}{2d+2}}$ where d in the dimension of the state space and K is the number of episodes. Lakshmanan et al. [2015] improve the latter work by considering a kernel density estimator instead of a frequency estimator for the transition probabilities. They achieve a regret bound of $K^{\frac{d+1}{d+2}}$. Yang et al. [2019] consider a deterministic control system under a Lipschitz assumption of the optimal action-value functions and the transition function and they establish a regret of $K^{\frac{d-1}{d}}$ where d here is the doubling dimension. Recently, Song and Sun [2019] extended the tabular Q-learning with upper-confidence bound exploration strategy, developed in Jin et al. [2018], to continuous state-action MDPs using a uniform discretisation of the joint space leading to the regret bound $\tilde{O}(H^{\frac{5}{2}}K^{\frac{d+1}{d+2}})$ where H is the planning horizon and d is the covering dimension. They only assume that the optimal action-value function is Lipschitz continuous. This assumption is more general than that used in the aforementioned works as it is known that Lipschitz continuity of the reward function and

the transition kernel leads to Lipschitz continuity of the optimal action-value function [Asadi et al., 2018]. We use the same condition in this present paper.

Adaptive discretization: Our method is closely related to methods that learn partition from continuous bandit literature [Kleinberg et al., 2008, Bubeck et al., 2009, Slivkins, 2014, Azar et al., 2014, Munos et al., 2014]. In particular, our method is inspired by the contextual Zooming algorithm introduced in Slivkins [2014] for contextual bandits, that we extend in non-trivial way to episodic RL setting. Our method is similar to two recently proposed algorithms. Zhu and Dunson [2019] propose and analyze an adaptive partitioning algorithm approach in the specific case where the metric space is a subset of $\mathbb{R}^d$ equipped with l_∞ distance as similarity metric. Concurrently to our work, Sinclair et al. [2019] extend the latter result to any generic metric space. However, their algorithm ADAPTIVE Q-LEARNING requires, at each re-partition step, a packing oracle that is able to take a region and value r and outputs an r-packing of that region. Whereas, our algorithm is oracle-free and creates at most a single subregion when needed. More comparison with this work requires the introduction of some notations and is therefore deferred to Section 5.5.

5.4 PROBLEM STATEMENT

5.4.1 Episodic Reinforcement Learning and Regret

We consider a finite horizon MDP $(\mathcal{S}, \mathcal{A}, P, r, H)$. We assume that reward $r_h(s,a) \in [0,1]$ for any state-action (s,a) and $h \in [H]$. We recall that for any step $h \in [H]$ and $(s,a) \in \mathcal{S} \times \mathcal{A}$, the state-action value function of a non-stationary deterministic policy $\pi = (\pi_1, \ldots, \pi_H)$ is defined as $Q_h^\pi(s,a) = r_h(s,a) + \mathbb{E}\left[\sum_{i=h+1}^H r_i(s_i, \pi_i(s_i)) \,\middle|\, s_h = s, a_h = a\right]$, and the value function is $V_h^\pi(s) = Q_h^\pi(s, \pi_h(s))$. As the horizon is finite, under some regularity conditions [Shreve and Bertsekas, 1978], there always exists an optimal policy $\pi^\star$ whose value and action-value functions are defined as $V_h^\star(x) \triangleq V_h^{\pi^\star}(s) = \max_\pi V_h^\pi(s)$ and $Q_h^\star(s,a) \triangleq Q_h^{\pi^\star}(s,a) = \max_\pi Q_h^\pi(s,a)$. Moreover, both Q^π and $Q^\star$ can be conveniently written as the result of the following Bellman equations

$$Q_h^\pi(s,a) = r_h(s,a) + (P_h V_{h+1}^\pi)(s,a), \tag{5.4.1}$$
$$Q_h^\star(s,a) = r_h(s,a) + (P_h V_{h+1}^\star)(s,a), \tag{5.4.2}$$

where $V_{H+1}^\pi(s) = V_{H+1}^\star(s) = 0$ and $V_h^\star(s) = \max_{a \in \mathcal{A}} Q_h^\star(s,a)$, for all $s \in \mathcal{S}$.

We focus on the online episodic reinforcement learning setting in which the reward and the transition kernel are unknown. The learning agent plays the game for K episodes $k = 1, \ldots, K$, where each episode k starts from some initial state s_1^k sampled according to some initial distribution. The agent controls the system by choosing a policy π_k at the beginning of the k-th episode. The total expected regret is defined then

$$\textsc{Regret}(K) \triangleq \sum_{k=1}^{K} V_1^*(s_1^k) - V_1^{\pi_k}(s_1^k).$$

5.4.2 Metric Space

We assume that the state-action space $\mathcal{X} \triangleq \mathcal{S} \times \mathcal{A}$ is compact endowed with a metric $\texttt{dist} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$. This leads us to state our main assumption:

Assumption 4 (Lipschitz Continuous $Q^\star$)**.** *We assume that for any $h \in [H]$, $Q_h^\star$ is L-Lipschitz continuous: for all $(s,a),(s',a') \in \mathcal{S} \times \mathcal{A}$*

$$|Q_h^\star(s,a) - Q_h^\star(s',a')| \leq L \cdot \texttt{dist}\left((s,a),(s',a')\right),$$

and without loss of generality,

$$\texttt{dist}\left((s,a),(s',a')\right) \leq 1, \forall (s,a),(s',a') \in \mathcal{S} \times \mathcal{A}.$$

Assumption 4 tells us that the optimal action values of nearby state-action pairs are close.

For a metric space $\mathcal{X}$ and $\epsilon > 0$, we denote the ϵ-net, $\mathcal{N}(\epsilon) \subset \mathcal{X}$, as a set such that

$$\forall x \in \mathcal{X}, \quad \exists x' \in \mathcal{N}(\epsilon), \quad \texttt{dist}(x,x') \leq \epsilon.$$

If $\mathcal{X}$ is compact, we denote $N(\epsilon)$ as the minimum size of an ϵ-net for $\mathcal{X}$. The covering dimension d of $\mathcal{X}$ is defined

$$d \triangleq \inf_{d'} \{d' \geq 0, \forall \epsilon > 0 \quad N(\epsilon) \leq \epsilon^{-d'}\}.$$

In particular, if $\mathcal{X}$ is a subset of Euclidean space equipped with l_p distance then its covering dimension is at most the linear dimension of $\mathcal{X}$. In many applications of interests, state-action spaces are commonly thought to be concentrated near a lower-dimensional manifold lying in high-dimensional ambient space. In this case, the covering dimension is much smaller than the linear dimension of the ambient space.

Covering is closely related to packing. We denote an ϵ-packing, $\mathcal{M}(\epsilon) \subset \mathcal{X}$, as a set such that

$$\forall x, x' \in \mathcal{M}(\epsilon), \quad \texttt{dist}(x,x') > \epsilon.$$

If $\mathcal{X}$ is compact, we denote $M(\epsilon)$ as the maximum size of an ϵ-packing. $N(\epsilon)$ and $M(\epsilon)$ have the same scaling as we have $M(2\epsilon) \leq N(\epsilon) \leq M(\epsilon)$.

5.5 THE ZOOMRL ALGORITHM

The ZOOMRL algorithm, shown in Algorithm 5, incrementally builds an optimistic estimate of the optimal action-value function over $\mathcal{X}$. The main idea is to estimate Q-values precisely in near-optimal regions, while estimating it loosely in sub-optimal regions. To implement this idea, we learn a partition of the space by zooming in more promising and frequently visited regions.

ZOOMRL maintains a partition of the space $\mathcal{X}$ that consists of a growing set of balls, of various sizes. Initially the set contains a single ball which includes the entire state-action space. Over time the set is expanded to include additional balls. The algorithm assigns two quantities to each ball: the number of times the ball is selected and an optimistic estimate of the Q-value of its center. By interpolating between these estimates using the Lipschitz structure, the algorithm assigns a tighter upper bound (called *index*) of the Q-value of each ball's center. These *indices* are then used to select the next ball and the next action to execute (line 8 -10 of Algorithm 5). Based on the received reward and the observed next state, the algorithm updates the selected ball's statistics (cf line 14-17 of Algorithm 5). Then, one ball may be created inside the selected ball according to an *activation rule* that reflects a bias-variance tradeoff (line 19-22 of Algorithm 5).

We denote by $\mathcal{B}_h$ the set of balls at step $h \in [H]$ that may change from episode to episode. Each ball $B = \{x \in \mathcal{X}, \texttt{dist}(x_B, x) \leq \texttt{rad}(B)\}$ has a radius $\texttt{rad}(B)$, center $x_B = (s_B, a_B)$ and a domain. The domain of ball $B \in \mathcal{B}_h$, denoted by $\texttt{dom}_h(B)$, is defined as the subset of B that excludes all active balls in $\mathcal{B}_h$ that have radius strictly smaller than $\texttt{rad}(B)$, i.e

$$\texttt{dom}_h(B) \triangleq B \setminus \left(\bigcup_{\substack{B' \in \mathcal{B}_h \\ \texttt{rad}(B') < \texttt{rad}(B)}} B' \right).$$

B is called *relevant* to a state s at step h if $(s, a) \in \texttt{dom}_h(B)$ for some action $a \in \mathcal{A}$. We denote the set of relevant balls to a given state s at step h by $\texttt{rel}_h(s) \triangleq \{B \in \mathcal{B}_h : \exists a \in \mathcal{A}, (s, a) \in \texttt{dom}_h(B)\}$. For each ball $B \in \mathcal{B}_h$ for some h, we keep track of the number of times B is selected at step h (denoted by $n_h(B)$), as well as a high probability upper bound (denoted by $\widehat{\mathrm{Q}}_h(B)$) for the optimal Q-value of the center of B (i.e $Q^\star_h(s_B, a_B)$).

Using the Lipschitz continuity assumption, we have that $L \cdot \texttt{rad}(B) + \widehat{\mathrm{Q}}_h(B') + L \cdot \texttt{dist}(x_B, x_{B'})$ is a valid high probability upper bound on $Q^\star(s_B, a_B)$ for any $B' \in \mathcal{B}_h$. Consequently, we get a tighter (less overoptimistic) upper bound, denoted by $\texttt{index}_h(B)$, by taking the minimum of these bounds

$$\texttt{index}_h(B) \triangleq L \cdot \texttt{rad}(B) + \min_{\substack{B' \in \mathcal{B}_h \\ \texttt{rad}(B') \geq \texttt{rad}(B)}} \{\widehat{\mathrm{Q}}_h(B') + L \cdot \texttt{dist}(B, B')\},$$

where, by abuse of notation, we write $\texttt{dist}(B, B') = \texttt{dist}(x_B, x_{B'})$.

To facilitate the algorithm's description, we introduce episode-indexed versions of the quantities, as shown in algorithm 5. We will use s_h^k, a_h^k and B_h^k to represent the state, the action and the ball generated at time step h of the k-th episode. Moreover, $\widehat{\mathrm{Q}}_h^k(B)$ and $n_h^k(B)$ are the statistics associated with each ball B at time step h at the beginning of the k-th episode.

The algorithm proceeds as follows. Initially, ZOOMRL creates a ball centered at arbitrary state-action pair with radius 1, hence covering the whole space. At step h of the k-th episode, a state s_h^k is observed, the algorithm finds the set of relevant balls to s_h^k (i.e $\texttt{rel}_h^k(s_h^k)$) and picks the ball B_h^k with the largest index (i.e $\texttt{index}_h^k$) among the relevant balls. Once the ball is selected, an action a_h^k is chosen randomly among actions a satisfying $(s_h^k, a) \in \texttt{dom}_h^k(B_h^k)$. Action a_h^k is then executed in the environment, a reward r_h^k is obtained and next state s_{h+1}^k is observed.

Based on the received reward and next state, the algorithm updates the statistics of the selected ball. The number of visits $n_h^k(B_h^k)$ is incremented by 1: $n_h^{k+1}(B_h^k) = n_h^k(B_h^k) + 1$. Let $t = n_h^{k+1}(B_h^k)$, the Q-value estimate is updated as follows

$$\widehat{\mathrm{Q}}_h^{k+1}(B_h^k) \leftarrow (1-\alpha_t)\,\widehat{\mathrm{Q}}_h^k(B_h^k) + \alpha_t\big(r_h^k + \widehat{\mathrm{V}}_{h+1}^k(s_{h+1}^k) + u_t + 2L \cdot \texttt{rad}(B_h^k)\big).$$

$V_{h+1}^k(x_{h+1}^k) = \min\{H, \max_{B \in \texttt{rel}_{h+1}^k(s_{h+1}^k)} \texttt{index}_{h+1}^k(B)\}$ here is the estimate of the next state's value and $\alpha_t \triangleq \frac{H+1}{H+t}$ is a learning rate. The term $u_t + 2L \cdot \texttt{rad}(B_h^k)$ corresponds to an exploration bonus used to bound estimation errors on the value function with high probability. The first term of the bonus is set to $u_t = 4\sqrt{\frac{H^3\iota}{t}}$ (we use $\iota \triangleq \log(4HK^2/p)$ for $p \in (0,1)$ to denote the log factor). It corresponds to a Hoeffding-style bonus which reflects the sample uncertainty due to insufficient number of samples. The second term $2L \cdot \texttt{rad}(B_h^k)$ accounts for the maximum possible variation of Q-values over the selected ball B_h^k.

Contrary to Q-values, the value of next state $\widehat{\mathrm{V}}_h^k(s_{h+1}^k)$ is defined over the entire state space, we don't need to maintain $\widehat{\mathrm{V}}_h$ but we query it whenever we need. In particular, $\widehat{\mathrm{V}}_h^k(s_{h+1}^k)$ is defined by the largest index among the relevant balls to s_{h+1}^k clipped above by H. The clipping here is to keep the value estimate into the range of plausible values while preserving the optimism as H is an upper bound on the true optimal value function.

Finally, ZOOMRL may create a new ball according to the following *activation rule*: If $n_h^k(B_h^k) \geq \frac{1}{\texttt{rad}(B_h^k)^2}$, a new ball B', centered in (s_h^k, a_h^k) and radius $\texttt{rad}(B') = \texttt{rad}(B_h^k)/2$, is created and its Q value is initialized to H. The activation criterion (which is equivalent to $\frac{1}{\sqrt{n_h^k(B_h^k)}} \leq \texttt{rad}(B_h^k)$) reflects a tradeoff

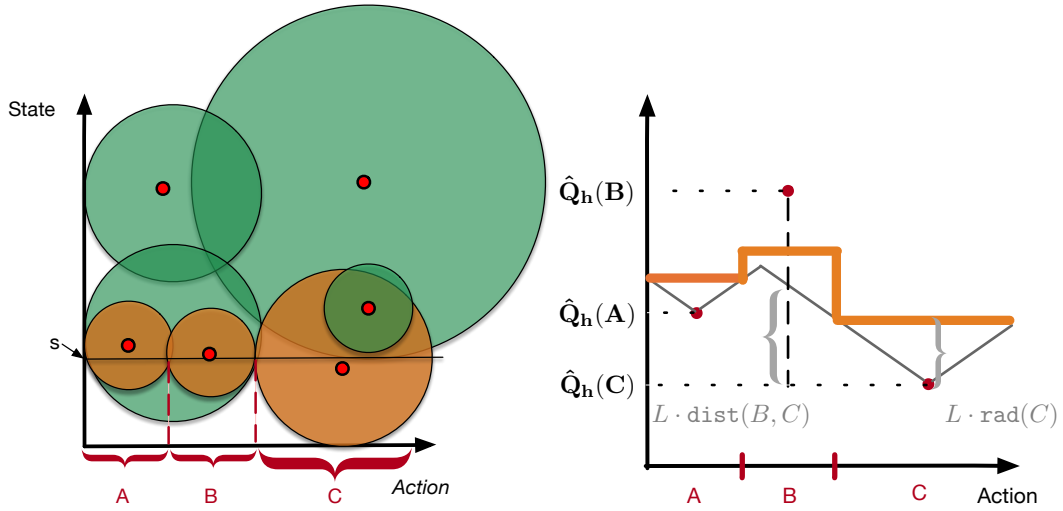

Figure 5.1: Left: a possible partition for 2-dimensional state-action space. For a given state s, we show in orange the three relevant balls A, B and C. Right: For the three relevant balls, we show how the `index` is constructed based on the interpolation between Q-value estimates of each ball. The gray piecewise linear curve corresponds to the function : $a \to \min_{B'}\{\widehat{Q}_h(B') + L \cdot \texttt{dist}((s,a), x_{B'})\}$ for a given state.

between the variance of Q-value estimate, due to the number of samples (i.e $\frac{1}{\sqrt{n_h^k(B)}}$) and the bias, corresponding to the radius of the ball.

Comparison with Sinclair et al. [2019]: Concurrently to our work, Sinclair et al. [2019] use a similar approach to learn an adaptive discretization. We highlight here differences between ZOOMRL and their algorithm ADAPTIVE Q-LEARNING:

- ADAPTIVE Q-LEARNING requires, at each re-partition step, a packing oracle that is able to take a ball B and value r to output an r-packing of B. Whereas, our algorithm is oracle-free and creates at most a single child ball when needed.

- We leverage more the Lipschitz structure to define the ball's *index*, which is not used in their algorithm.

- Sinclair et al. [2019] use an exploration bonus $2\sqrt{\frac{H^3 \log(4HK/p)}{t}} + \frac{4L}{\sqrt{t}}$ where $t = n_h^k(B_h^k)$. The first term looks similar to our term u_t with K^2 instead of K in the log factor. We think it is due to small issue in their proof because there is a missing union bound over K possible values of the random stop-

ping time $t = n_h^k(B_h^k)$ (cf Proof C.1.3 in appendix). The second term, $\frac{4L}{\sqrt{t}}$, is different than ours, $L \cdot \texttt{rad}(B_h^k)$.

- Finally, in [Sinclair et al., 2019] each child ball inherents statistics from their parent while in our algorithms the statistics are initialized by zero for n_h and H for $\widehat{\mathrm{Q}}_h$.

Algorithm 5 ZOOMRL

Data: For $h \in [H]$, we have a collection $\mathcal{B}_h$ of balls.
Init: create ball B, with $\texttt{rad}(B) = 1$ and arbitrary center. $\mathcal{B}_h^1 \leftarrow \{B\}$ for all $h \in [H]$
$\widehat{\mathrm{Q}}_h^1(B) = H$ and $n_h^1(B) = 0, \forall h \in [H]$
for episode $k = 1, \ldots, K$ **do**
  Observe x_1^k
  for step $h = 1, \ldots, H$ **do**
    // Select action
    $B_h^k \leftarrow \arg\max_{B \in \texttt{rel}_h^k(s_h^k)} \texttt{index}_h^k(B)$
    $a_h^k \leftarrow$ any arm a such that $(s_h^k, a) \in \mathrm{dom}(B_h^k)$
    Execute action a_h^k, observe reward r_h^k and next state s_{h+1}^k
    // Query the next value function
    $\widehat{\mathrm{V}}_{h+1}^k(s_{h+1}^k) \leftarrow \min\{H, \max_{B \in \texttt{rel}_{h+1}^k(s_{h+1}^k)} \texttt{index}_{h+1}^k(B)\}$
    // Update the selected ball's statistics
    $t = n_h^{k+1}(B_h^k) \leftarrow n_h^k(B_h^k) + 1$
    $u_t \leftarrow 4\sqrt{\frac{H^3 \iota}{t}}$
    $\widehat{\mathrm{Q}}_h^{k+1}(B_h^k) \leftarrow (1-\alpha_t)\,\widehat{\mathrm{Q}}_h^k(B_h^k) + \alpha_t \left(r_h^k + \widehat{\mathrm{V}}_{h+1}^k(s_{h+1}^k) + u_t + 2L \cdot \texttt{rad}(B_h^k)\right)$
    // New ball's activation step
    if $n_h^k(B) \geq \frac{1}{\texttt{rad}(B_h^k)^2}$ **then**
      Create a new ball B' centered in (s_h^k, a_h^k) and radius $\texttt{rad}(B') = \texttt{rad}(B_h^k)/2$
      $\mathcal{B}_h^{k+1} = \mathcal{B}_h^k \cup B'$
      $\widehat{\mathrm{Q}}_h^{k+1}(B') = H$ and $n_h^{k+1}(B') = 0, \forall h \in [H]$
    end if
  end for
end for

5.6 MAIN RESULTS

In this section, we present our main theoretical result which is an upper bound on the total regret of ZOOMRL (see Algorithm 5). We start by showing a pessimistic version of the regret bound.

Theorem 1 (Worst case guarantee). *For any $p \in (0,1)$, with probability $1-p$, the total regret of* ZOOMRL *(see Algorithm 5) is at most $O(\sqrt{H^5 \iota} L K^{\frac{d+1}{d+2}})$ where $\iota = \log(4HK^2/p)$ and d is the covering dimension of the state-action space.*

The bound in Theorem 1 matches the regret bound achieved by Net-based Q-learning (NBQL) studied in Song and Sun [2019] which assumes access to an ϵ-net of the whole space as input to the algorithm. Moreover, the ϵ-net should be optimal in the sense that the granularity ϵ of the covering must be chosen in advance ($\epsilon = K^{\frac{-1}{d+2}}$). Meanwhile, ZOOMRL builds the partition on the fly and in data-dependent fashion by allocating more effort in promising regions, which would considerably save the memory requirement in favorable problems while preserving the worst-case guarantee (as shown in Theorem 1).

Now, we present a refined regret bound that reflects better the geometry of the underlying space.

Theorem 2 (Refined regret bound). *For any $p \in (0,1)$, with probability $1-p$, the total regret of* ZOOMRL *(see Algorithm 5) is at most*

$$O\left((L + \sqrt{H^5 \iota}) \min_{r_0 \in (0,1)} \left\{ K r_0 + \sum_{\substack{r = 2^{-i} \\ r \geq r_0}} \frac{M(r)}{r} \right\} + H^2 + \sqrt{H^3 K \iota} \right),$$

where $\iota = \log(4HK^2/p)$, $M(r)$ is the r-packing number of the state-action space.

Since $M(r)$ is non-increasing in r, the leading term (the first term) of the bound of Theorem 2 is upper bounded by $\min_{r_0 \in (0,1)} \left\{ K r_0 + \frac{M(r_0)}{r_0} \log(r_0) \right\}$. By setting $r_0 = K^{\frac{-1}{d+2}}$, we recover the worst-case bound in Theorem 1. Note that the work of Sinclair et al. [2019] achieves the same regret bound.

The r-packing number $M(r)$ is here to uniformly upper bound the number of balls of radius r generated by the algorithm, as we will see in the analysis deferred to the next section. Intuitively, balls with small radius would not cover the whole state-action space but rather would be concentrated around near-optimal regions. We expect that their number would be much smaller that $M(r)$ in practice.

Comparison with contextual bandit setting: We would like to highlight a negative result of the RL setting comparing to the contextual bandit setting. When $H = 1$ and if we ignore logarithmic factors in Theorem 2, we obtain a bound in $\min_{r_0 \in (0,1)} \left\{ K r_0 + \sum_{r=2^{-i} \geq r_0} \frac{M(r)}{r} \right\}$. This looks similar to the regret bound of the contextual Zooming algorithm [Slivkins, 2014]. But there is a crucial difference: $M(r)$ here is the r-packing of the entire space while it is replaced by the r-packing of near-optimal regions in Slivkins [2014]. This follows from the fact that in contextual bandit setting, total regret could be straightforwardly written as sum of instant regrets incurred by each ball. Such regret is bounded, up to a multiplicative constant, by the radius of the ball in which the context falls.

However the dependence of our regret bound on K is still optimal, up to logarithmic factor, with respect to the worst Lipschitz structure. In fact, theorem 8 in Slivkins [2014] states that there exists a distribution $\mathcal{I}$ over problem instances on $(\mathcal{S} \times \mathcal{A}, \texttt{dist})$ such that for any contextual bandit algorithm, the expected regret over $\mathcal{I}$ is lower bounded by $\Omega \left(\min_{r_0 \in (0,1)} \left\{ K r_0 + \sum_{r=2^{-i} \geq r_0} \frac{M(r)}{r} \right\} / \log(K) \right)$.

Tabular MDP In the case of finite state-action MDP without any structural knowledge, one can pick the metric to be $\texttt{dist}((s,a),(s',a')) = H, \quad \forall (s,a) \neq (s',a')$. It is obvious that the optimal action-value function is 1-Lipschitz (i.e $L = 1$) with respect to this metric and that the packing number is at most equal to $|\mathcal{S}||\mathcal{A}|$. In this case, if we set r_0 to $\sqrt{\frac{|\mathcal{S}||\mathcal{A}|}{K}}$, the regret bound in Theorem 2 becomes $O(\sqrt{|\mathcal{S}||\mathcal{A}|H^5 K\imath})$. Hence, we recover exactly the regret bound of Q-learning with UCB-Hoeffding algorithm of Jin et al. [2018].

5.6.1 Result For The Misspecified Case

We study now how ZoomRL deals with misspecification error. First, we present a formal definition for an approximate Lipscthiz Q-value.

Assumption 5 (Approximately Lipschitz $Q^\star$)**.** *We assume that for any $h \in [H]$, $Q_h^\star$ can be decomposed as a L-Lipschitz continuous term and a bounded term as:*

$$\forall (s,a) \in \mathcal{X}, Q_h^\star(s,a) = f_h(s,a) + \Delta_h(s,a),$$

where for all $(s,a),(s',a') \in \mathcal{X}$

$$|f_h(s,a) - f_h(s',a')| \leq L \cdot \texttt{dist}\left((s,a),(s',a')\right) \text{ and}$$
$$|\Delta_h(s,a)| \leq \epsilon.$$

A straightforward consequence of Assumption 4 is:

$$|Q_h^\star(s,a) - Q_h^\star(s',a')| \leq L \cdot \texttt{dist}((s,a),(s',a')) + 2\epsilon.$$

The next theorem states that our algorithm, without any modification, is robust to small misspecification error ϵ.

Theorem 3 (Regret bound in the misspecified case)**.** *Suppose that Assumption 5 holds. For any $p \in (0,1)$, with probability $1-p$, the total regret of* ZoomRL *(see Algorithm 5) is at most*

$$O\left((L + \sqrt{H^5 \iota}) \min_{r_0 \in (0,1)} \left\{ K r_0 + \sum_{\substack{r = 2^{-i} \\ r \geq r_0}} \frac{M(r)}{r} \right\} + H^2 + \sqrt{H^3 K \iota} + HK\epsilon \right),$$

where $\iota = \log(4HK^2/p)$, $M(r)$ is the r-packing number of the state-action space.

The Theorem 3 states that ZoomRL incurs at most an extra regret term $O(HK\epsilon)$, comparing to Theorem 2. This term is linear in the number of episodes K as well as the error ϵ. The good news is that our algorithm, without any adaptation, does not break down entirely and it enjoys good guarantees when the optimal Q-value is close to a Lipschitz function i.e the error ϵ is small.

5.7 Proof Outline

In this section we outline some key steps in the proof of Theorem 2. All the omitted proofs as well as the analysis of the misspecified case can be found in the appendix. We start by showing two useful properties of our partitioning scheme.

Lemma 3 (Partition's properties)**.** *At each step $h \in [H]$ in episode $k \in [k]$, we have*

- *The domains of balls cover the space-action space i.e $\cup_{B \in \mathcal{B}_h^k} \texttt{dom}_h^k(B) = \mathcal{S} \times \mathcal{A}$.*
- *For any two balls of radius $r > 0$, their centers are at distance at least r. In other words, for any $r > 0$, the set $\{B \in \mathcal{B}_h^k, \texttt{rad}(B) = r\}$ forms an r-packing for $\mathcal{S} \times \mathcal{A}$.*

We set $\alpha_t = \frac{H+1}{H+t}$. This specific choice of learning rate comes from Jin et al. [2018] where they show that this choice is crucial to obtain regret that is not exponential in H. We denote $\alpha_t^0 = \prod_{j=1}^t (1-\alpha_j)$ and $\alpha_t^i = \alpha_i \prod_{j=i+1}^t (1-\alpha_j)$. We have $\alpha_t^0 = 0, \forall t \geq 1$ and $\alpha_t^0 = 1$ when $t = 0$. The lemma below establishes the recursive formula of Q-values estimates for the balls.

Lemma 4. *At any $(h,k) \in \times[H] \times [K]$ and $B \in \mathcal{B}_h^k$, let $t = n_h^k(B)$, and suppose B was previously selected at step h of episodes $k_1, k_2, ..., k_t < k$. By the update rule of $\widehat{\mathrm{Q}}$, we have:*

$$\widehat{\mathrm{Q}}_h^k(B) = \alpha_t^0 \cdot H + \sum_{i=1}^{t} \alpha_t^i \cdot \Big(r_h(x_h^{k_i}, a_h^{k_i}) + \widehat{\mathrm{V}}_{h+1}^{k_i}(x_{h+1}^{k_i}) + u_i + 2L \cdot \mathtt{rad}(B) \Big).$$

Throughout the learning process, we hope that our estimation $\widehat{\mathrm{Q}}_h^k$ will get closer to the optimal value $Q_h^\star$, as k increases while we preserve optimism. Using Azuma-Hoeffding concentration inequality (see Lemma 15 in appendix) together with the Lipschitz assumption, our next lemma shows that $\widehat{\mathrm{Q}}_h^k$ is an upper bound on $Q_h^\star$ at any episode k with high probability and the difference between $\widehat{\mathrm{Q}}_h^k$ and $Q_h^\star$ is controlled by quantities from the next step.

Lemma 5. *For any $p \in (0,1)$, we have $\beta_t = 2\sum_{i=1}^t \alpha_t^i u_i \le 16\sqrt{\frac{H^3\iota}{t}}$ and, with probability at least $1 - p/2$, we have that for all $(s,a,h,k) \in \mathcal{S} \times \mathcal{A} \times [H] \times [K]$ and any ball $B \in \mathcal{B}_h^k$ such that $(s,a) \in dom_h^k(B)$:*

(a) $\widehat{\mathrm{Q}}_h^k(B) \ge Q_h^\star(s,a)$.

(b) $\widehat{\mathrm{Q}}_h^k(B) - Q_h^\star(x,a) \le \alpha_t^0 \cdot H + \beta_t + 4L \cdot \mathtt{rad}(B) + \sum_{i=1}^t \alpha_t^i \cdot (\widehat{\mathrm{V}}_{h+1}^{k_i} - V_{h+1}^*)(s_{h+1}^{k_i})$.

where $t = n_h^k(B)$ and $k_1, \cdots, k_t < k$ are the episodes where B was selected at step h.

The next lemma translates the optimism in terms of Q-value estimates to optimism in terms of value function estimates.

Lemma 6 (Optimism)**.** *Following the same setting as in Lemma 5, for any (h,k), with probability at least $1 - p/2$, we have for any $s \in \mathcal{S}$, $\widehat{\mathrm{V}}_h^k(s) \ge V_h^\star(s)$.*

Proof. Let $s \in \mathcal{S}$, We have $V_h^\star(s) = Q_h^\star(s, \pi_h^\star(s))$. As the set of domains of active balls covers the entire space, there exists $B^\star \in \mathcal{B}_{h+1}^k$ such that $(s, \pi_h^\star(s)) \in \mathtt{dom}_h^k(B^*)$. By the definition of index, we have $\mathtt{index}_h^k(B^\star) = L \cdot \mathtt{rad}(B^\star) +$

$\widehat{\mathrm{Q}}_h^k(\tilde{B}^\star) + L \cdot \mathtt{dist}(\tilde{B}^\star, B^\star)$ for some active ball $\tilde{B}^\star$. We have

$$
\begin{aligned}
&\widehat{\mathrm{V}}_h^k(s) - V_h^\star(s) \\
&= \min\{H, \max_{B \in \mathtt{rel}_h^k(s)} \mathtt{index}_h^k(B)\} - Q_h^\star(s, \pi^\star(s)) \\
&\geq \max_{B \in \mathtt{rel}_h^k(s)} \mathtt{index}_h^k(B) - Q_h^\star(s, \pi^\star(s)) \\
&\geq \mathtt{index}_h^k(B^\star) - Q_h^\star(s, \pi^\star(s)) \\
&= L \cdot \mathtt{rad}(B^\star) + \widehat{\mathrm{Q}}_h^k(\tilde{B}^\star) + L \cdot \mathtt{dist}(\tilde{B}^\star, B^\star) - Q_h^\star(s, \pi^\star(s)) \\
&\geq L \cdot \mathtt{rad}(B^\star) + Q_h^\star(s_{\tilde{B}^\star}, a_{\tilde{B}^\star}) + L \cdot \mathtt{dist}(\tilde{B}^\star, B^\star) - Q_h^\star(s, \pi^\star(s)) \\
&\geq L \cdot \mathtt{rad}(B^\star) + Q_h^\star(s_{B^\star}, a_{B^\star}) - Q_h^\star(s, \pi^\star(s)) \\
&\geq 0
\end{aligned}
$$

Where $(s_{B^\star}, a_{B^\star})$ and $(s_{\tilde{B}^\star}, a_{\tilde{B}^\star})$ denote respectively the centers of balls $B^\star$ and $\tilde{B}^\star$. The first inequality follows from $Q_h^\star(s,a) \leq H$ for any state-action pair (s,a). The third inequality follows from lemma 5. The fourth and the last inequalities follow from Lipschitz assumption 4 □

5.7.1 Regret Analysis

π_k is the policy executed by the algorithm in step h for H steps to reach the end of the episode. By the optimism of our estimates with respect to the true value function (see lemma 6), we have with probability at least $1 - p/2$

$$
\textsc{Regret}(K) = \sum_{k=1}^{K} (V_1^* - V_1^{\pi_k})(x_1^k) \leq \sum_{k=1}^{K} (\widehat{\mathrm{V}}_1^k - V_1^{\pi_k})(x_1^k)
$$

Denote by $\delta_h^k \triangleq (\widehat{\mathrm{V}}_h^k - V_h^{\pi_k})(s_h^k)$ and $\phi_h^k \triangleq (\widehat{\mathrm{V}}_h^k - V_h^\star)(s_h^k)$. As $V_h^\star \geq V^{\pi_k}$, we have $\phi_h^k \leq \delta_h^k$. In the sequel, we aim to upper bound δ_h^k as we have $\textsc{Regret}(K) \leq \sum_{k=1} \delta_1^k$.

Let B_h^k the ball selected at step h of episode k and $B_{\mathtt{init}}$ be the initial ball of radius one that covers the whole space. We denote $B_h^{k,\mathtt{pa}}$ the parent of B_h^k. When B_h^k is the initial ball, we consider that it is parent of itself.

Lemma 7 (Bound on estimation)**.** *Let $\hat{P}_h$ the empirical transition operator defined as $[\hat{P}_h V](s_h^k, a_h^k) = V(s_{h+1}^k)$ for any function V, any h and k. If we denote $\xi_{h+1}^k =$*

$[(P_h - \hat{P}_h)(V^\star_{h+1} - V^{\pi_k}_h)](s^k_h, a^k_h)$, *we have*

$$\delta^k_h \leq H\alpha^0_{n^k_h(B^k_h)} \cdot \mathbb{I}_{\{B^k_h = B_{init}\}} + (11L + 32\sqrt{H^3\iota})\, \mathtt{rad}(B^k_h) + \sum_{i=1}^{n^k_h(B^{k,pa}_h)} \alpha^i_{n^k_h(B^{k,pa}_h)} \phi^{k_i(B^{k,pa}_h)}_{h+1} - \phi^k_{h+1} + \delta^k_{h+1} + \xi^k_{h+1},$$

where $k_i(B^{k,pa}_h)$ *is the i-th episode where* $B^{k,pa}_h$ *is selected at step h.*

Taking the sum over $k \in [K]$ of the estimation bound in lemma 7,

$$\sum_{k=1}^K \delta^k_h \leq \square + (11L + 32\sqrt{H^3\iota}) \sum_{k=1}^K \mathtt{rad}(B^k_h) + \triangle + \sum_{k=1}^K (-\phi^k_{h+1} + \delta^k_{h+1} + \xi^k_{h+1}),$$

where $\square = H\sum_{k=1}^K \alpha^0_{n^k_h(B^k_h)} \cdot \mathbb{I}_{\{B^k_h = B_{\mathtt{init}}\}}$ and $\triangle = \sum_{k=1}^K \sum_{i=1}^{n^k_h(B^{k,\mathtt{pa}}_h)} \alpha^i_{n^k_h(B^{k,\mathtt{pa}}_h)} \phi^{k_i(B^{k,\mathtt{pa}}_h)}_{h+1}$. For the fist term, we have $\square = H\sum_{k=1}^K \mathbb{I}_{\{B^k_h = B_{\text{init}}, n^k_h(B_{\text{init}})=0\}} = H$. For the second term $\triangle$, we regroup the summation in a different way. For every $k' \in [K]$, the term $\phi^{k'}_{h+1}$ appears in the summation with $k > k'$ when B^k_h and $B^{k'}_h$ share the same parent. The first time it appears we have $n^k_h(B^{k,\mathtt{pa}}_h) = n^{k'}_h(B^{k',\mathtt{pa}}_h) + 1$, the second time it appears we have $n^k_h(B^{k,\mathtt{pa}}_h) = n^{k'}_h(B^{k',\mathtt{pa}}_h) + 2$ and so on. Therefore:

$$\begin{aligned}\triangle &\leq \sum_{k'=1}^K \phi^{k'}_{h+1} \sum_{t=n^{k'}_h(B^{k',\mathtt{pa}}_h)+1}^{\infty} \alpha_t^{n^{k'}_h(B^{k',\mathtt{pa}}_h)} \\ &\leq \left(1 + \frac{1}{H}\right) \sum_{k=1}^K \phi^k_{h+1}.\end{aligned}$$

We use in the last inequality $\sum_{t=i}^\infty \alpha^i_t = 1 + \frac{1}{H}$ (Lemma 16 in appendix). Therefore, using that $\phi^k_{h+1} \leq \delta^k_{h+1}$

$$\begin{aligned}\sum_{k=1}^K \delta^k_h &\leq H + (11L + 32\sqrt{H^3\iota}) \sum_{k=1}^K \mathtt{rad}(B^k_h) \\ &\quad + \left(1 + \frac{1}{H}\right) \sum_{k=1}^K \phi^k_{h+1} + \sum_{k=1}^K (-\phi^k_{h+1} + \delta^k_{h+1} + \xi^k_{h+1}) \\ &\leq H + (11L + 32\sqrt{H^3\iota}) \sum_{k=1}^K \mathtt{rad}(B^k_h) + \left(1 + \frac{1}{H}\right) \sum_{k=1}^K \delta^k_{h+1} + \sum_{k=1}^K \xi^k_{h+1}.\end{aligned}$$

By unrolling the last inequality for $h \in [H]$ and using the fact $\delta_{H+1}^k = 0 \quad \forall k \in [K]$, we obtain

$$\sum_{k=1}^{K} \delta_1^k \leq \sum_{h=1}^{H} (1 + \frac{1}{H})^{h-1} \left(H + (11L + 32\sqrt{H^3\iota}) \sum_{k=1}^{K} \texttt{rad}(B_h^k) + \sum_{k=1}^{K} \xi_{h+1}^k \right)$$
$$\leq 3H^2 + 3(11L + 32\sqrt{H^3\iota}) \sum_{h=1}^{H} \sum_{k=1}^{K} \texttt{rad}(B_h^k) + 3 \sum_{h=1}^{H} \sum_{k=1}^{K} \xi_{h+1}^k. \tag{5.7.1}$$

The last inequality follows from the fact $\forall h \in [H], \left(1 + \frac{1}{H}\right)^{h-1} \leq \left(1 + \frac{1}{H}\right)^{H} \leq \exp(1) \leq 3$.

Now, we proceed to upper bound the two terms $\sum_{h=1}^{H} \sum_{k=1}^{K} \texttt{rad}(B_h^k)$ and $\sum_{h=1}^{H} \sum_{k=1}^{K} \xi_{h+1}^k$. Using concentration argument, we show with probability at least $1 - p/2$, we have (see Lemma 12 in appendix)

$$\sum_{h=1}^{H} \sum_{k=1}^{K} \xi_{h+1}^k \leq 4\sqrt{2H^3K\iota} \tag{5.7.2}$$

Bounding $\sum_{h=1}^{H} \sum_{k=1}^{K} \texttt{rad}(B_h^k)$: Let's consider all balls of radius r that have been activated at step h throughout the execution of the algorithm. The maximum number of times a ball B of radius r can be selected before it becomes a parent is upper bounded by $\frac{1}{r^2}$. After ball B becomes a parent, a new ball of radius $r/2$ is created every time B is selected. Therefore, we can write the sum over all ball $B \in \mathcal{B}_h^K$ of radius r as the sum over set of rounds which consists of the round when B was created and all rounds when B was selected before being

a parent. Let $r_0 \in (0,1)$. we have

$$\begin{aligned}
\sum_{k=1}^{K} \texttt{rad}(B_h^k) &= \sum_{r=2^{-i}} \sum_{\substack{B\in\mathcal{B}_h^K \\ \texttt{rad}(B)=r}} \sum_{\substack{k\in[K] \\ B_h^k=B}} r \\
&= \sum_{\substack{r=2^{-i} \\ r<r_0}} \sum_{\substack{B\in\mathcal{B}_h^K \\ \texttt{rad}(B)=r}} \sum_{\substack{k\in[K] \\ B_h^k=B}} r + \sum_{\substack{r=2^{-i} \\ r\geq r_0}} \sum_{\substack{B\in\mathcal{B}_h^K \\ \texttt{rad}(B)=r}} \sum_{\substack{k\in[K] \\ B_h^k=B}} r \\
&\leq Kr_0 + \sum_{\substack{r=2^{-i} \\ r\geq r_0}} \sum_{\substack{B\in\mathcal{B}_h^K \\ \texttt{rad}(B)=r}} \frac{r}{r^2} + 2r \\
&\leq Kr_0 + 3 \sum_{\substack{r=2^{-i} \\ r\geq r_0}} |\{B \in \mathcal{B}_h^K : \texttt{rad}(B) = r\}| \frac{1}{r} \\
&\leq Kr_0 + 3 \sum_{\substack{r=2^{-i} \\ r\geq r_0}} \frac{M(r)}{r} \qquad (5.7.3)
\end{aligned}$$

The last step follows from lemma 3: The set of active balls of radius r is a r-packing of $\mathcal{S}\times\mathcal{A}$. Thus, $|\{B \in \mathcal{B}_h^K : \texttt{rad}(B) = r\}| \leq M(r)$ where $M(r)$ is the r-covering number.

Plugging bounds (5.7.2) and (5.7.3) in (5.7.1) and using the union bound, we obtain the desired regret bound in Theorem 2

5.8 Conclusion

In this paper, we present ZoomRL, a provably efficient model-free reinforcement learning algorithm in continuous state-action spaces under the assumption that the true optimal action-value function is Lipscthiz with respect to similarity metric between state-action pairs. Our algorithm takes into account the geometry of the action-value function by allocating more attention to relevant regions. We show that our method achieves sublinear regret that depends on the packing number of the state-action space and that it is robust to small misspecification errors.

Our method requires the knowledge of the Lipschitz constant L as well as the metric `dist` to achieve its performance. A natural future question is whether an RL algorithm can be proved to be efficient without knowing L or `dist` in advance.

6 Online Learning in Non-stationary Linear Markov Decision Processes

6.1 Prologue to the Contribution

6.1.1 Article Details

This chapter is based on the article «*Efficient Learning in Non-Stationary Linear Markov Decision Processes*» [Touati and Vincent, 2020a], joint work with Pascal Vincent. I am the first author. I performed the derivation of the new algorithm and its theoretical analysis.

6.1.2 Context

In order to design both provable sample-efficient and computationally efficient RL algorithms for large-scale problems, an appealing challenge is to combine exploration strategies with generalization methods. In the previous chapter, we ensured generalization over states by aggregating them into a finite set of meta-states and run tabular exploration mechanism on the latter. In particular, we proposed to actively explore the state-action space by learning on-the-fly an adaptive partitioning that takes into account the shape of the optimal value function. When the state-action space is assumed to be a compact metric space, we showed that such adaptive discretization-based algorithm yield sublinear regret but suffer from the curse of dimensionality as their regret scales almost exponentially with the covering dimension of the whole space.

Another structural assumption, that received attention in the recent literature [Yang and Wang, 2019, Jin et al., 2020b, Zanette et al., 2020a], is when both reward and transition dynamics are linear functions with respect to a given feature mapping. This assumption enables the design of efficient algorithms with a linear representation of the action-value function.

Most prior algorithms with linear function approximation assume that the environment is stationary and minimize the regret over the best fixed policy. However, in many problems of interest, we are faced with a changing world, in some cases with substantial non-stationarity. This is a more realistic and challenging setting, since what has been learned in the past may be obsolete in the present.

6.1.3 Paper Abstract

We study episodic reinforcement learning in non-stationary linear (a.k.a. low-rank) Markov Decision Processes (MDPs), *i.e*, both the reward and transition kernel are linear with respect to a given feature map and are allowed to evolve either slowly or abruptly over time. For this problem setting, we propose OPT-WLSVI an optimistic model-free algorithm based on weighted least squares value iteration which uses exponential weights to smoothly forget data that are far in the past. We show that our algorithm, when competing against the best policy at each time, achieves a regret that is upper bounded by $\widetilde{\mathcal{O}}(d^{5/4}H^2\Delta^{1/4}K^{3/4})$ where d is the dimension of the feature space, H is the planning horizon, K is the number of episodes and Δ is a suitable measure of non-stationarity of the MDP. Moreover, we point out technical gaps in the study of forgetting strategies in non-stationary linear bandits setting made by previous works and we propose a fix to their regret analysis.

6.1.4 Recent Developments

The authors of Cheung et al. [2019] who pioneered the non-stationary linear bandit, released recently a revised version of their AISTATS 2019 paper to acknowledge the mistake in the analysis. In order for their optimal rate $\tilde{\mathcal{O}}(T^{2/3})$ to hold, they assume that actions are orthogonal.

Zhao and Zhang [2021] identify also the mistake and proposed the same fix than ours in a technical note released slightly after we had made public on arxiv a version of our paper containing the fix.

Faury et al. [2021] build on our proposed fix to provide a correct regret analysis for non-stationary generalized linear bandit.

While our work shows that forgetting strategies achieve the rate of $\tilde{\mathcal{O}}(T^{3/4})$, the recent work Wei and Luo [2021] follows a substantially different approach to achieve the optimal rate $\tilde{\mathcal{O}}(T^{2/3})$ for the first time in the setting of non-stationary linear bandit and MDP. Their algorithm detects non-stationarity by running multiple instances of a base (stationary) algorithm with different durations in a randomized schedule.

6.2 Introduction

In the present work, we study the problem of online learning in episodic non-stationary linear Markov Decision Processes (MDP), where both the reward and transition kernel are linear with respect to a given feature map and are allowed to evolve dynamically and even adversarially over time. The interaction of the

agent with the environment is divided into K episodes of fixed length H. Moreover, we assume that the total change of the MDP, that we measure by a suitable metric, over the K episodes is upper bounded by Δ, called *variation budget*.

To address this problem, we propose a computationally efficient model-free algorithm, that we call OPT-WLSVI. We prove that, in the setting described above, its regret when competing against the best policy for each episode is at most $\widetilde{\mathcal{O}}(d^{5/4}H^2\Delta^{1/4}K^{3/4})$. Concurrently to our work, Zhou et al. [2020] propose to periodically restart LSVI-UCB from scratch, achieving the same regret. By contrast, our algorithm is based on weighted least squares value iteration that uses exponential weights to smoothly forget data that are far in the past, which drives the agent to keep exploring to discover changes. Our approach is motivated by the recent work of Russac et al. [2019] who establish a new deviation inequality to sequential weighted least squares estimator and apply it to the non-stationary stochastic linear bandit problem. However, in contrast to linear bandit, our algorithm handles the additional problem of credit assignment since future states depend in non-trivial way on the agent's policy and thus we need to carefully control how errors are propagated through iterations. Moreover, we discovered technical errors in the regret analysis of forgetting strategies in non-stationary linear bandits made by previous works and we propose a correction.

6.3 Problem Statement

6.3.1 Notation

Throughout the paper, all vectors are column vectors. We denote by $\|\cdot\|$ the Euclidean norm for vectors and the operator norm for matrices. For positive definite matrix A, we use $\|x\|_A$ to denote the matrix norm $\sqrt{x^\top Ax}$. We define $[N]$ to be the set $\{1, 2, \dots, N\}$ for any positive integer N.

6.3.2 Non-Stationary Reinforcement Learning and Dynamic Regret

We consider a non-stationary undiscounted finite-horizon MDP $(\mathcal{S}, \mathcal{A}, P, r, H)$ where $\mathcal{S}$ and $\mathcal{A}$ are the state and action space, H is the planning horizon i.e number of steps in each episode, $P = \{P_{t,h}\}_{t>0,h\in[H]}$ and $r = \{r_{t,h}\}_{t>0,h\in[H]}$ are collections of transition kernels and reward functions, respectively. More precisely, when taking action a in state s at step h of the t-th episode, the agent receives a reward $r_{t,h}(s,a)$ and makes a transition to the next state according to the probability measure $P_{t,h}(\cdot \mid s,a)$.

For any step $h \in [H]$ of an episode t and $(s,a) \in \mathcal{S} \times \mathcal{A}$, the state-action value function of a policy $\pi = (\pi_1, \ldots, \pi_H)$ is defined as $Q^{\pi}_{t,h}(s,a) = r_{t,h}(s,a) + \mathbb{E}\left[\sum_{i=h+1}^{H} r_{t,i}(s_i, \pi_i(s_i)) \,\middle|\, s_h = s, a_h = a\right]$, and the value function is $V^{\pi}_{t,h}(s) = Q^{\pi}_{t,h}(s, \pi_h(s))$. The optimal value and action-value functions are defined as $V^{\star}_{t,h}(x) \triangleq \max_{\pi} V^{\pi}_{t,h}(s)$ and $Q^{\star}_{t,h}(s,a) \triangleq \max_{\pi} Q^{\pi}_{t,h}(s,a)$. If we denote $[P_{t,h}V_{t,h+1}](s,a) = \mathbb{E}_{s' \sim P_{t,h}(\cdot|s,a)}[V_{t,h+1}(s')]$, both Q^{π} and $Q^{\star}$ can be conveniently written as the result of the following Bellman equations

$$Q^{\pi}_{t,h}(s,a) = r_{t,h}(s,a) + [P_{t,h}V^{\pi}_{t,h+1}](s,a), \tag{6.3.1}$$
$$Q^{\star}_{t,h}(s,a) = r_{t,h}(s,a) + [P_{t,h}V^{\star}_{t,h+1}](s,a), \tag{6.3.2}$$

where $V^{\pi}_{t,H+1}(s) = V^{\star}_{t,H+1}(s) = 0$ and $V^{\star}_{t,h}(s) = \max_{a \in \mathcal{A}} Q^{\star}_{t,h}(s,a)$, for all $s \in \mathcal{S}$.

Learning problem: We focus on the online episodic reinforcement learning setting in which the rewards and the transition kernels are unknown. The learning agent plays the game for K episodes $t = 1, \ldots, K$, where each episode t starts from some initial state $s_{t,1}$ sampled according to some initial distribution. The agent controls the system by choosing a policy π_t at the beginning of the t-th episode. We measure the agent's performance by the dynamic regret, defined as the sum over all episodes of the difference between the optimal value function in episode t and the value of π_t:

$$\textsc{Regret}(K) = \sum_{t=1}^{K} V^{*}_{t,1}(s_{t,1}) - V^{\pi_t}_{t,1}(s_{t,1}).$$

6.3.3 Linear Markov Decision Processes

In this work, we consider a special class of MDPs called linear MDPs, where both reward function and transition kernel can be represented as a linear function of a given feature mapping $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$. Now we present our main assumption

Assumption 6 (Non-stationary linear MDP)**.** *$(\mathcal{S}, \mathcal{A}, P, r, H)$ is non-stationary linear MDP with a feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ if for any $(t,h) \in \mathbb{N} \times [H]$, there exist d unknown (signed) measures $\boldsymbol{\mu}_{t,h} = (\mu^{(1)}_{t,h}, \ldots, \mu^{(d)}_{t,h})$ over $\mathcal{S}$ and an unknown vector $\boldsymbol{\theta}_{t,h} \in \mathbb{R}^d$, such that for any $(s,a) \in \mathcal{S} \times \mathcal{A}$, we have*

$$P_{t,h}(\cdot \mid s,a) = \boldsymbol{\phi}(s,a)^{\top} \boldsymbol{\mu}_{t,h}(\cdot), \tag{6.3.3}$$
$$r_{t,h}(s,a) = \boldsymbol{\phi}(s,a)^{\top} \boldsymbol{\theta}_{t,h}. \tag{6.3.4}$$

Without loss of generality, we assume[1] $\|\phi(s,a)\| \leq 1$ *for all* $(s,a) \in \mathcal{S} \times \mathcal{A}$, *and* $\max\{\|\boldsymbol{\mu}_{t,h}(\mathcal{S})\|, \|\boldsymbol{\theta}_{t,h}\|\} \leq \sqrt{d}$ *for all* $(t,h) \in \mathbb{N} \times [H]$.

Linear MDPs are also know as low-rank MDPs [Zanette et al., 2020a]. In fact, in the case of finite state and action spaces with cardinalities $|\mathcal{S}|$ and $|\mathcal{A}|$ respectively, the transition matrix $P \in \mathbb{R}^{(|\mathcal{S}|\times|\mathcal{A}|)\times|\mathcal{S}|}$ could be expressed by the following low-rank factorization for any (t,h):

$$P_{t,h} = \boldsymbol{\Phi} \boldsymbol{M}_{t,h}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{(|\mathcal{S}|\times|\mathcal{A}|)\times d}$ such as $\boldsymbol{\Phi}[(s,a),:] = \phi(s,a)^\top$ and $\boldsymbol{M}_{t,h} \in \mathbb{R}^{d\times|\mathcal{S}|}$ such that $\boldsymbol{M}_{t,h}[:,s] = \boldsymbol{\mu}_{t,h}(s)$ a discrete measure. Therefore the rank of the matrix P is at most d.

An important consequence of Assumption 6 is that the Q-function of any policy is linear in the features ϕ.

Lemma 8. *For every policy* π *and any* $(t,h) \in \mathbb{N}^\star \times [H]$ *there exists* $\mathbf{w}^\pi_{t,h} \in \mathbb{R}^d$ *such that*

$$Q^\pi_{t,h} = \phi(s,a)^\top \mathbf{w}^\pi_{t,h}, \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}. \tag{6.3.5}$$

6.4 The Proposed Algorithm

Algorithm 6, referred as OPT-WLSVI (OPTimistic Weighted Least Squares Value Iteration), parametrizes the Q-values $Q_{t,h}(s,a)$ by a linear form $\phi(s,a)^\top \mathbf{w}_{t,h}$ and updates the parameters $\mathbf{w}_{t,h}$ by solving the following regularized weighted least squares problem:

$$\mathbf{w}_{t,h} = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \left\{ \sum_{\tau=1}^{t-1} \eta^{-\tau} \left(r_{\tau,h} + V_{t,h+1}(s_{\tau,h+1}) - \phi^\top_{\tau,h}\mathbf{w} \right)^2 + \lambda \eta^{-(t-1)} \|\mathbf{w}\|^2 \right\}$$

where $\eta \in (0,1)$ is a discount factor, $V_{t,h+1}(s_{t,h+1}) = \max_{a\in\mathcal{A}} Q_{t,h+1}(s_{\tau,h+1},a)$ and $r_{\tau,h}$ and $\phi_{\tau,h}$ are shorthand for $r_{\tau,h}(s_{\tau,h},a_{\tau,h})$ and $\phi(s_{\tau,h},a_{\tau,h})$ respectively. The discount factor η plays an important role as it gives exponentially increasing weights to recent transitions, hence, the past is smoothly forgotten.

The regularized weighted least-squares estimator of the above problem can be written in closed form

$$\mathbf{w}_{t,h} = \Sigma^{-1}_{t,h} \left(\sum_{\tau=1}^{t-1} \eta^{-\tau} \phi_{\tau,h} (r_{\tau,h} + V_{t,h+1}(s_{\tau,h+1})) \right) \tag{6.4.1}$$

[1] A concrete case that would satisfy these assumptions, is if $\forall i \in [d], \phi_i(s,a) \geq 0$ and $\sum_{i=1}^d \phi_i(s,a) = 1$, and $\forall i \in [d], \boldsymbol{\mu}^{(i)}_{t,h}$ is a probability measure. In this case ϕ(s,a) can be understood as providing the mixture coefficients with which to mix the d measures in $\boldsymbol{\mu}_{t,h}$.

where $\Sigma_{t,h} = \sum_{\tau=1}^{t-1} \eta^{-\tau} \phi_{\tau,h} \phi_{\tau,h}^\top + \lambda \eta^{-(t-1)} \cdot \mathbf{I}$ is the Gram matrix. We further define the matrix

$$\widetilde{\Sigma}_{t,h} = \sum_{\tau=1}^{t-1} \eta^{-2\tau} \phi_{\tau,h} \phi_{\tau,h}^\top + \lambda \eta^{-2(t-1)} \cdot \mathbf{I} \tag{6.4.2}$$

The matrix $\widetilde{\Sigma}_{t,h}$ is connected to the variance of the estimator $w_{t,h}$, which involves the squares of the weights $\{\eta^{-2\tau}\}_{\tau \geq 0}$. OPT-WLSVI uses both matrices $\Sigma_{t,h}$ and $\widetilde{\Sigma}_{t,h}$ to define a upper confidence bound (UCB) term $\beta(\phi^\top \Sigma_{t,h}^{-1} \widetilde{\Sigma}_{t,h} \Sigma_{t,h}^{-1} \phi)^{1/2}$ to encourage exploration, where β is a scalar.

The algorithm proceeds as follows. At the beginning of episode t, OPT-WLSVI estimates the weighted least square estimator $w_{t,h}$ for each step $h \in [H]$ as given by Equation (6.4.1). Then, the algorithm updates the Q-value and the value function estimates as follows:

$$\begin{aligned} Q_{t,h}(\cdot, \cdot) &= \phi(\cdot, \cdot)^\top \mathbf{w}_{t,h} + \beta(\phi(\cdot, \cdot)^\top \Sigma_{t,h}^{-1} \widetilde{\Sigma}_{t,h} \Sigma_{t,h}^{-1} \phi(\cdot, \cdot))^{1/2} \\ V_{t,h}(\cdot) &= \min\{\max_{a \in \mathcal{A}} Q_{t,h}(\cdot, s), H\} \end{aligned}$$

The UCB term is used to bound the estimation error of the value function, due to an insufficient number of samples, with high probability. The clipping of the value estimate is here to keep $V_{t,h}$ within the range of plausible values while preserving the optimism as H is an upper bound on the true optimal value function. Finally the algorithm collects a new trajectory by following the greedy policy π_t with respect to the estimated Q-values.

Computational complexity: At each step $h \in [H]$ of an episode $t \in [K]$, we need to compute the inverse of $\Sigma_{t,h}$ to solve the weighted least-squares problem. A naive implementation requires $\mathcal{O}(d^3)$ elementary operations, but as $\Sigma_{t,h}$ is essentially a sum of rank-one matrices, we need only $\mathcal{O}(d^2)$ using the Sherman-Morrison update formula. Furthermore, $\mathcal{O}(d^2)$ operations are needed to compute the exploration bonus $(\phi^\top \Sigma_{t,h}^{-1} \widetilde{\Sigma}_{t,h} \Sigma_{t,h}^{-1} \phi)^{1/2}$ that can be computed using only matrix-vector multiplications. Therefore computing $V_{t,h+1}$ for all the past successor states requires $\mathcal{O}(d^2|\mathcal{A}|K)$ (the $|\mathcal{A}|$ factor is due to the maximization over actions). As we need to do this at all steps and for every episode, the overall computation complexity of our algorithm is $\mathcal{O}(d^2|\mathcal{A}|HK^2)$.

6.5 NON-STATIONARY LINEAR BANDITS

Before providing the analysis of OPT-WLSVI, we start by examining the linear bandit case when the horizon $H = 1$. Let us first recall the non-stationary linear bandit model

Algorithm 6 Optimistic Weighted Least-Squares Value Iteration (OPT-WLSVI)

for episode $t = 1, \dots, K$ **do**
  Receive the initial state $s_{t,1}$.
  /* Run LSVI procedure
  $V_{t,H+1}(\cdot) \leftarrow 0$
  for step $h = H, \dots, 1$ **do**
    $\mathbf{w}_{t,h} \leftarrow \Sigma_{t,h}^{-1}(\sum_{\tau=1}^{k-1} \eta^{-\tau} \phi_{\tau,h}(r_{\tau,h} + V_{t,h+1}(s_{\tau,h+1})))$
    $Q_{t,h}(\cdot,\cdot) \leftarrow \phi(\cdot,\cdot)^\top \mathbf{w}_{t,h} + \beta(\phi(\cdot,\cdot)^\top \Sigma_{t,h}^{-1} \widetilde{\Sigma}_{t,h} \Sigma_{t,h}^{-1} \phi(\cdot,\cdot))^{1/2}$
    $V_{t,h}(\cdot) \leftarrow \min\{\max_{a \in \mathcal{A}} Q_{t,h}(\cdot, s), H\}$
  end for
  end for
  /* Execute greedy policy
  for step $h = 1, \dots, H$ **do**
    Execute $a_{t,h} = \operatorname{argmax}_{a \in \mathcal{A}} Q_{t,h}(s_{t,h}, a)$
    receive $r_{t,h}$ and observe $s_{t,h+1}$
    /* Update matrices
    $\Sigma_{t+1,h} \leftarrow \Sigma_{t,h} + \eta^{-t} \phi_{t,h} \phi_{t,h}^\top + \lambda \eta^{-t}(1-\eta) \cdot \mathbf{I}$
    $\widetilde{\Sigma}_{t+1,h} \leftarrow \widetilde{\Sigma}_{t,h} + \eta^{-2t} \phi_{t,h} \phi_{t,h}^\top + \lambda \eta^{-2t}(1-\eta^2) \cdot \mathbf{I}$
  end for
end for
end for

Definition 1 (Non-stationary linear bandit)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ a set of decisions. At iteration t, the player makes a decision x_t from a subset set $\mathcal{X}_t \subset \mathcal{X}$, then observes the reward r_t satisfying:*

$$r_t = x_t^\top \boldsymbol{\theta}_t + z_t \tag{6.5.1}$$

where $\boldsymbol{\theta}_t$ is the unknown regression parameter at iteration t and z_t is conditionally σ-subgaussian noise. We assume further that $\|x\| \leq 1, \forall x \in \mathcal{X}$ and $\|\boldsymbol{\theta}_t\| \leq S, \forall t$.

When $H = 1$ linear MDP reduces to linear bandit if we let $\mathcal{X} = \{\phi(s,a), a \in \mathcal{A}, s \in \mathcal{S}\}$ and $\mathcal{X}_t = \{\phi(s_t, a), a \in \mathcal{A}\}$ where s_t is sampled from a given fixed distribution over states.

For the bandit setting, forgetting strategies have been proposed such as sliding-window, weighted regression and restarting in [Cheung et al., 2019], [Russac et al., 2019] and Zhao et al. [2020] respectively. Randomized exploration with weighting strategy has also been introduced in Kim and Tewari [2020]. The aforementioned works provide a regret of $\tilde{\mathcal{O}}(d^{2/3}\Delta^{1/2}K^{2/3})$ which is optimal as it matches the established lower bound $\Omega(d^{2/3}\Delta^{1/3}K^{2/3})$ in [Cheung et al., 2019] up to $\log(K)$ factors. Unfortunately, we find technical gaps in the regret analysis provided by the earliest paper [Cheung

et al., 2019], which were then reproduced by the other three papers. Specifically Cheung et al. [2019] attempted, in their Lemma 1, to upper bound the non-stationarity bias of the reward parameters by controlling the eigenvalues of matrix $M = V_t^{-1} \sum_{\tau=t-W}^{p} x_\tau x_\tau^\top$, where $V_t = \sum_{\tau=1}^{t-1} x_\tau x_\tau^\top + \lambda \cdot \mathbf{I}$ is the Gram matrix and for any integer $p \in \{t-W, \dots, t-1\}$. They then needed to prove that M is positive semi-definite, but their argument has technical errors. We precise in the appendix the issue in their argument and we provide concrete counter-examples.

Now, we provide a fix to the original error in the regret analysis of SW-UCB algorithm in Cheung et al. [2019] (see also Appendix D.2 for the analysis of D-LINUCB algorithm proposed by Russac et al. [2019]). At time t, SW-UCB selects a decision as follows:

$$x_t = \arg\max_{x \in \mathcal{X}_t} x^\top \hat{\boldsymbol{\theta}}_t + \beta \|x\|_{V_t^{-1}} \tag{6.5.2}$$

where $\hat{\boldsymbol{\theta}}_t = V_t^{-1} \sum_{\tau=\max\{1,t-W\}}^{t-1} x_\tau r_\tau$ is the solution of the sliding window least squares problem

In their Lemma 1, Cheung et al. [2019] attempts to control the non-stationarity bias $\|\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}}_t\|$ where $\bar{\boldsymbol{\theta}}_t \triangleq V_t^{-1} \sum_{\tau=\max\{1,t-W\}}^{t-1} A_\tau A_\tau^\top \boldsymbol{\theta}_\tau + \lambda \boldsymbol{\theta}_t$ is the average of the true regression parameters over the sliding window. We propose to control $|x^\top(\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}}_t)|$ for any $x \in \mathcal{X}$ and then use the fact that $\|\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}}_t\| = \max_{x:\|x\|=1} |x^\top(\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}}_t)|$

$$\begin{aligned}
&|x^\top(\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}}_t)| \\
&= \left| x^\top V_t^{-1} \sum_{\tau=\max\{1,t-W\}}^{t-1} x_\tau x_\tau^\top (\boldsymbol{\theta}_\tau - \boldsymbol{\theta}_t) \right| \\
&\leq \sum_{\tau=\max\{1,t-W\}}^{t-1} \left| x^\top V_t^{-1} x_\tau \right| \cdot |x_\tau^\top (\boldsymbol{\theta}_\tau - \boldsymbol{\theta}_t)| && \text{(triangle inequality)} \\
&= \sum_{\tau=\max\{1,t-W\}}^{t-1} |x^\top V_t^{-1} x_\tau| \cdot |x_\tau^\top (\sum_{s=\tau}^{t-1} (\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}))| \\
&\leq \sum_{\tau=\max\{1,t-W\}}^{t-1} |x^\top V_t^{-1} x_\tau| \cdot \|x_\tau\| \cdot \| \sum_{s=\tau}^{t-1} (\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}) \| && \text{(Cauchy-Schwarz)} \\
&\leq \sum_{\tau=\max\{1,t-W\}}^{t-1} |x^\top V_t^{-1} x_\tau| \cdot \sum_{s=\tau}^{t-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\| && (\|x_\tau\| \leq 1)
\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{s=\max\{1,t-W\}}^{t-1} \sum_{\tau=\max\{1,t-W\}}^{s} |x^\top V_t^{-1} x_\tau| \cdot \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\| \\
&\qquad\qquad (\textstyle\sum_{\tau=\max\{1,t-W\}}^{t-1} \sum_{s=\tau}^{t-1} = \sum_{s=\max\{1,t-W\}}^{t-1} \sum_{\tau=\max\{1,t-W\}}^{s}) \\
&\leq \sum_{s=\max\{1,t-W\}}^{t-1} \sqrt{\left[\sum_{\tau=\max\{1,t-W\}}^{s} x^\top V_t^{-1} x\right] \cdot \left[\sum_{\tau=\max\{1,t-W\}}^{s} x_\tau^\top V_t^{-1} x_\tau\right]} \cdot \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\| \\
&\qquad\qquad \text{(Cauchy-Schwarz)} \\
&\leq \sum_{s=\max\{1,t-W\}}^{t-1} \sqrt{\left[\sum_{\tau=\max\{1,t-W\}}^{s} x^\top V_t^{-1} x\right] \cdot d} \cdot \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\| \qquad ((\star)) \\
&\leq \|x\| \sqrt{d} \sum_{s=\max\{1,t-W\}}^{t-1} \sqrt{\frac{\sum_{\tau=\max\{1,t-W\}}^{t-1} 1}{\lambda}} \cdot \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\| \qquad (\lambda_{\max}(V_t^{-1}) \leq \tfrac{1}{\lambda}) \\
&\leq \|x\| \sqrt{\frac{dW}{\lambda}} \sum_{s=\max\{1,t-W\}}^{t-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|
\end{aligned}
$$

where the inequality $(\star)$ follows from the fact that $\sum_{\tau=\max\{1,t-W\}}^{s} x_\tau^\top V_t^{-1} x_\tau \leq d$ that can be proved as follows. We have $\sum_{\tau=\max\{1,t-W\}}^{t-1} x_\tau^\top V_t^{-1} x_\tau = \sum_{\tau=\max\{1,t-W\}}^{t-1} \mathrm{tr}\left(x_\tau^\top V_t^{-1} x_\tau\right) = \mathrm{tr}\left(V_t^{-1} \sum_{\tau=\max\{1,t-W\}}^{t-1} x_\tau x_\tau^\top\right)$. Given the eigenvalue decomposition $\sum_{\tau=\max\{1,t-W\}}^{t-1} x_\tau x_\tau = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)^\top$, we have $V_t = \mathrm{diag}(\lambda_1 + \lambda, \ldots, \lambda_d + \lambda)^\top$, and $\mathrm{tr}\left(V_t^{-1} \sum_{\tau=1}^{t-1} x_\tau x_\tau^\top\right) = \sum_{i=1}^{d} \frac{\lambda_j}{\lambda_j + \lambda} \leq d$.

Comparing to the bound on $\|\bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t\|$ in the Lemma 1 of Cheung et al. [2019], there is an extra factor $\sqrt{\frac{dW}{\lambda}}$ that multiplies the local non-stationarity term $\sum_{s=t-W}^{t-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|$. This extra factor will consequently multiply the variation budget term in the final regret, as stated in the following proposition:

Proposition 5. *Under the assumption that* $\sum_{t=1}^{K-1} \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t+1}\| \leq \Delta$, *for any* $\delta \in (0, 1)$, *if we set* $\beta = \sqrt{\lambda} S + \sigma \sqrt{2 \log(K/\delta) + d \log(1 + \frac{W}{\lambda d})}$ *in the algorithm 1* SW-UCB *of Cheung et al. [2019], then with probability* $1 - \delta$, *the dynamic regret of* SW-UCB *is at most*

$$
\mathcal{O}\left(\sqrt{\frac{dW}{\lambda}} \Delta W + \beta \sqrt{dK} \sqrt{\lceil K/W \rceil} \sqrt{\log(1 + \frac{W}{d\lambda})}\right)
$$

Comparing to the regret upper bound in Theorem 3 of Cheung et al. [2019], our fix leads to an extra factor $\sqrt{dW}$ multiplying the variation budget in , which becomes now $\tilde{\mathcal{O}}(d^{1/2} \Delta W^{3/2} + dKW^{-1/2})$. Optimizing over the sliding window

size W leads to a final dynamic regret of $\tilde{\mathcal{O}}(d^{7/8}\Delta^{1/4}K^{3/4})$. Note that the latter is not optimal since it does not match the lower bound $\Omega(d^{2/3}\Delta^{1/3}K^{2/3})$. This leaves the question of whether or not forgetting strategies are optimal to handle non-stationarity in linear bandits as an open research problem.

6.6 THEORETICAL GUARANTEE OF OPT-WLSVI

In this section, we present our main theoretical result which is an upper bound on the dynamic regret of OPT-WLSVI (see Algorithm 6). First, we quantify the variations on reward function and transition kernel over time in terms of their respective variation budgets Δ_r and Δ_P. The main advantage of using the variation budget is that it accounts for both slowly-varying and abruptly-changing MDPs.

Definition 2 (MDP Variation budget). *We define* $\Delta = \Delta_r + \Delta_P$ *where*

$$\Delta_r \triangleq \sum_{t=1}^{K}\sum_{h=1}^{H} \|\boldsymbol{\theta}_{t,h} - \boldsymbol{\theta}_{t+1,h}\|\,,$$
$$\Delta_P \triangleq \sum_{t=1}^{K}\sum_{h=1}^{H} \|\boldsymbol{\mu}_{t,h}(\mathcal{S}) - \boldsymbol{\mu}_{t+1,h}(\mathcal{S})\|\,.$$

A similar notion has already been proposed in the literature, for instance total variance distance between $P_{t,h}$ and $P_{t+1,h}$ in tabular MDPs [Ortner et al., 2019, Cheung et al., 2020] or Wasserstein distance in smooth MDPs [Domingues et al., 2020a].

Now we present our bound on the dynamic regret for OPT-WLSVI.

Theorem 4 (Regret Bound). *Under Assumption 6, there exists an absolute constant* $c > 0$ *such that, for any fixed* $\delta \in (0,1)$*, if we set* $\lambda = 1$ *and* $\beta = c \cdot dH\sqrt{\iota}$ *in Algorithm 6 with* $\iota \triangleq \log\left(\frac{2dH}{\delta(1-\eta)}\right)$*, then with probability* $1-\delta$*, for any* $W > 0$ *the dynamic regret of* OPT-WLSVI *is at most*

$$\mathcal{O}\Bigg(cd^{3/2}H\sqrt{K\iota}\sqrt{2K\log(1/\eta) + 2\log\left(1 + \frac{1}{d\lambda(1-\eta)}\right)} + H^{3/2}\sqrt{K\iota} + \underbrace{\sqrt{\frac{d}{\lambda(1-\eta)}}HW\Delta + \frac{H^2K\sqrt{d}}{\lambda}\frac{\eta^W}{1-\eta}}_{\textit{non-stationarity bias}}\Bigg), \tag{6.6.1}$$

where $\mathcal{O}(\cdot)$ *hides only absolute constants.*

Linear	Stationary	Non-stationary
Bandits	$\widetilde{\mathcal{O}}(dK^{1/2})$ [Abbasi-Yadkori et al., 2011]	$\widetilde{\mathcal{O}}(d^{7/8}\Delta^{1/4}K^{3/4})$ Cheung et al. [2019] Russac et al. [2019] and our work
MDPs	$\widetilde{\mathcal{O}}(d^{3/2}H^2K^{1/2})$ [Jin et al., 2020b] $\widetilde{\mathcal{O}}(dH^2K^{1/2})$ [Zanette et al., 2020b]	$\widetilde{\mathcal{O}}(d^{5/4}H^2\Delta^{1/4}K^{3/4})$ Our work

Table 6.1: Comparison of our regret bound with state-of-the-art bounds for both linear bandits and linear MDPs. d is the dimension of the features space, H is the planning horizon of the MDP, K is the number of episodes and Δ is the variation budget. When we go from a bandit setting to MDPs, the work of Jin et al. [2020b] in the stationary case and our work in the non-stationary case incur an extra $d^{1/2}$ factor and $d^{3/8}$ respectively. Zanette et al. [2020b] achieve a linear dependence on d in the stationary case but their proposed algorithm is computationally intractable.

The last two terms of the of Equation (6.6.1) are the result of the bias due to the non-stationarity of the MDP. In theorem 4 we introduce the parameter W that can be interpreted, at a high level, as the effective temporal window equivalent to a particular choice of discount factor η: the bias resulting from transitions that are within the window W may be bounded in term of W while the remaining ones are bounded globally by the last term of Equation (6.6.1).

The following corollary shows that by optimizing the parameters W and η, our algorithm achieves a sublinear regret.

Corollary 1. *If we set* $\log(1/\eta) = \left(\frac{\Delta}{dK}\right)^{1/2}$ *and* $W = \frac{\log(K/(1-\eta))}{\log(1/\eta)}$*; under the same assumptions as in Theorem 4, for any* $\delta \in (0,1)$*, we have that with probability* $1-\delta$*, the dynamic regret of* OPT-WLSVI *is at most* $\widetilde{\mathcal{O}}(d^{5/4}H^2\Delta^{1/4}K^{3/4})$ *where* $\widetilde{\mathcal{O}}(\cdot)$ *hides logarithmic factors.*

In Corollary 1, we rely on the knowledge of the variation budget Δ (or at least an upper bound on Δ) in order to achieve a sublinear regret. We show in the next section how to relax the requirement of knowing the variation budget. In particular, we will describe how to extend our algorithm, using the Bandit-over-Reinforcement-Learning framework [Cheung et al., 2020] in order to deal with an unknown variation budget.

6.6.1 Unknown variation budget

Our algorithm OPT-WLSVI needs the variation budget Δ to set the optimal value of the forgetting parameter as $\log(1/\eta^\star) = \left(\frac{\Delta}{dK}\right)^{1/2}$. We can use the Bandit-over-Reinforcement-Learning framework (BoRL) [Cheung et al., 2020] to tune the forgetting parameter online.

The idea is to run a multi-armed bandit algorithm over a set of sub-algorithm each using a different parameter. In our case, each sub-algorithm is a OPT-WLSVI with a different guess on $\eta^\star$. If $\Delta \geq \sqrt{K}$ the regret bound is vacuous (linear regret), we are only interested in problems with Δ in the range $[1, \sqrt{K}]$. This implies that the set of $\log(1/\eta)$ only needs to span the range $[\frac{1}{\sqrt{dK}}, \frac{1}{\sqrt{d}}]$.

We divide the horizon K into $\frac{K}{M}$ equal-length intervals each of length M, specified later. In each interval, sub-algorithm i restarts a OPT-WLSVI with $\log(1/\eta_i) = \frac{2^i}{\sqrt{dK}}$. We have in total $I = \lfloor \log_2(\sqrt{K}) \rfloor + 1$ possible value of $\log(1/\eta)$ in the form of $\frac{2^i}{\sqrt{dK}}$ that spans $[\frac{1}{\sqrt{dK}}, \frac{1}{\sqrt{d}}]$. We can verify that there exists $i^\star \in [A]$ such that $\log(1/\eta_{i^\star}) \leq \log(1/\eta^\star) \leq 2\log(1/\eta_{i^\star})$, which well-approximates the optimal parameter up to constant factors.

On top of these sub-algorithms, we run the EXP3.P [Auer et al., 2002]. The arms are the sub-algorithms. There are I arms and the reward for each arm or sub-algorithm i in interval $m \in [\frac{K}{M}]$ is the total of reward collected in the MDP during this interval. EXP3.P is called for $\frac{K}{M}$ rounds to select the arm.

Regret Analysis of OPT-WLSVI + BORL: Let i_m the arm selected by EXP3.P for the interval $m \in [\frac{K}{M}]$ and π^i is the algorithm followed by a sub-algorithm i. The regret of the overall algorithm can be decomposed as the regret of the algorithm $i^\star$ that optimally tunes the parameter plus the loss due to learning $i^\star$ with the EXP3.P algorithm:

$$\begin{aligned}\text{REGRET}(K) = &\left(\sum_{t=1}^{K} V_{t,1}^{\star}(s_t^1) - V_{t,1}^{\pi_t^{i^\star}}(s_t^1)\right) \\ &+ \left(\sum_{m=1}^{\frac{K}{M}} \sum_{t=(m-1)M+1}^{mM} V_{t,1}^{\pi_t^{i^\star}}(s_t^1) - V_{t,1}^{\pi_t^{i_m}}(s_t^1)\right)\end{aligned}$$

The fist term corresponds to OPT-WLSVI with parameter $\eta_{i^\star}$. Therefore, we can bound this term using Theorem 6.2. As $\eta_{i^\star}$ differs from $\eta^\star$ up to constant factor, we obtain the bound in Corollary 6.3 i.e $\tilde{\mathcal{O}}(d^{5/4}H^2\Delta^{1/4}K^{3/4})$.

The second term corresponds to the regret of the EXP3.P learner against the sub-algorithm $i^\star$. There are I arms, EXP3.P is called for $\frac{K}{M}$ rounds and the rewards collected during each interval is upperbounded by MH. Therefore, by a classical regret bound of EXP3.P [Auer et al., 2002], the second term is upper-bounded with high probability by:

$$\tilde{\mathcal{O}}(MH\sqrt{I\frac{K}{M}}) = \tilde{\mathcal{O}}(H\sqrt{MK})$$

We obtain that $\text{REGRET}(K) = \tilde{\mathcal{O}}(d^{5/4}H^2\Delta^{1/4}K^{3/4} + H\sqrt{MK})$ and by choosing $M = \sqrt{K}$, we conclude that $\text{REGRET}(K) = \tilde{\mathcal{O}}(d^{5/4}H^2\Delta^{1/4}K^{3/4})$. Note that we obtain the same regret bound when the variation budget is known.

6.7 Technical Highlights

In this section, we give an overview of some key ideas leading to the regret bound in Theorem 4. Inspired by the analysis of weighting approach in bandit [Russac et al., 2019], one can attempt to interpret the algorithm as acting optimistically with respect to the weighted parameters of the optimal Q-value defined as $\bar{\mathbf{w}}_{t,h}(s,a) = \Sigma_{t,h}^{-1}(\sum_{\tau=1}^{t-1}\eta^{-\tau}\boldsymbol{\phi}_{\tau,h}\boldsymbol{\phi}_{\tau,h}^{\top}\mathbf{w}_{\tau,h}^{\star} + \lambda\eta^{-(t-1)}\mathbf{w}_{t,h}^{\star})$, where $\mathbf{w}_{t,h}^{\star}$ are the true parameters of the optimal Q-value. This first attempt was unsuccessful. Then, we came up with the implicitly defined *weighted MDP* and we were able to interpret our algorithm as acting optimistically with respect to this weighted MDP. We provide the full proofs and derivations in the appendix. We first translate the parameter update produced by the algorithm into the following compact update of Q-value estimates for any $t \in [K]$ and $h \in \{H, \dots 1\}$:

$$Q_{t,h} = \widehat{r}_{t,h} + \widehat{P}_{t,h}V_{t,h+1} + B_{t,h} \tag{6.7.1}$$

where we define the implicitly empirical reward function $\widehat{r}$ and transition measure $\widehat{P}$ as follows:

$$\widehat{r}_{t,h}(s,a) \triangleq \boldsymbol{\phi}(s,a)^{\top}\Sigma_{t,h}^{-1}(\sum_{\tau=1}^{t-1}\eta^{-\tau}\boldsymbol{\phi}_{\tau,h}r_{\tau,h}),$$
$$\widehat{P}_{t,h}(\cdot \mid s,a) \triangleq \boldsymbol{\phi}(s,a)^{\top}\Sigma_{t,h}^{-1}(\sum_{\tau=1}^{t-1}\eta^{-\tau}\boldsymbol{\phi}_{\tau,h}\delta(\cdot, s_{\tau,h+1})),$$

and $B_{t,h}(\cdot,\cdot) = \beta(\boldsymbol{\phi}(\cdot,\cdot)^{\top}\Sigma_{t,h}^{-1}\widetilde{\Sigma}_{t,h}\Sigma_{t,h}^{-1}\boldsymbol{\phi}(\cdot,\cdot))^{1/2} = \beta\,\|\boldsymbol{\phi}(\cdot,\cdot)\|_{\Sigma_{t,h}^{-1}\widetilde{\Sigma}_{t,h}\Sigma_{t,h}^{-1}}$ is the exploration bonus.

We can interpret the Equation (6.7.1) as an approximation of the backward induction in a *weighted average* MDP defined formally as follows.

Definition 3 (Weighted Average MDP)**.** *let for any $(s,a) \in \mathcal{S} \times \mathcal{A}$,*

$$\bar{r}_{t,h}(s,a) \triangleq \boldsymbol{\phi}(s,a)^{\top}\Sigma_{t,h}^{-1}(\sum_{\tau=1}^{t-1}\eta^{-\tau}\boldsymbol{\phi}_{\tau,h}\boldsymbol{\phi}_{\tau,h}^{\top}\boldsymbol{\theta}_{\tau,h} + \lambda\eta^{-(t-1)}\boldsymbol{\theta}_{t,h}),$$
$$\bar{P}_{t,h}(\cdot \mid s,a) \triangleq \boldsymbol{\phi}(s,a)^{\top}\Sigma_{t,h}^{-1}(\sum_{\tau=1}^{t-1}\eta^{-\tau}\boldsymbol{\phi}_{\tau,h}\boldsymbol{\phi}_{\tau,h}^{\top}\boldsymbol{\mu}_{\tau,h}(\cdot) + \lambda\eta^{-(t-1)}\boldsymbol{\mu}_{t,h}(\cdot)).$$

$(\mathcal{S}, \mathcal{A}, \bar{P}, \bar{r})$ is called the weighted average MDP.

We can see that if we ignore the regularization term (we set λ to zero), $\widehat{r}_{t,h}$ coincides with $\bar{r}_{t,h}$ and $\widehat{P}_{t,h}$ is an unbiased estimate of $\bar{P}_{t,h}$. Therefore, in contrast with the stationary case, we are tracking the Q-value of the weighted average MDP instead of the true MDP at time t. The next Lemma quantifies the bias arising from the time variations of the environment.

Lemma 9 (Non-stationarity bias)**.** *For any $W \in [t-1]$ and for any bounded function $f : \mathcal{S} \to \mathbb{R}$ such as $\|f\|_\infty \leq H$, we have:*

$$|r_{t,h}(s,a) - \bar{r}_{t,h}(s,a)| \leq \texttt{bias}_r(t,h), \quad \left|[(P_{t,h} - \bar{P}_{t,h})f](s,a)\right| \leq H\texttt{bias}_P(t,h),$$

where

$$\texttt{bias}_r(t,h) = \sqrt{\frac{d}{\lambda(1-\eta)}} \sum_{s=t-W}^{t-1} \|\boldsymbol{\theta}_{s,h} - \boldsymbol{\theta}_{s+1,h}\| + \frac{2\sqrt{d}\eta^W}{\lambda(1-\eta)},$$
$$\texttt{bias}_P(t,h) = \sqrt{\frac{d}{\lambda(1-\eta)}} \sum_{s=t-W}^{t-1} \|\boldsymbol{\mu}_{s,h}(\mathcal{S}) - \boldsymbol{\mu}_{s+1,h}(\mathcal{S})\| + \frac{2\sqrt{d}\eta^W}{\lambda(1-\eta)}.$$

We analyse now the one-step error decomposition of the difference between the estimates $Q_{t,h}$ and $Q^\pi_{t,h}$ of a given policy π. To do that, we use the weighted MDP $(\mathcal{S}, \mathcal{A}, \bar{P}, \bar{r})$ to isolate the bias term. The decomposition contains four parts: the reward bias and variance, the transition bias and variance, and the difference in value functions at step $h+1$. It can be written as:

$$\phi(s,a)^\top \mathbf{w}_{t,h} - Q^\pi_{t,h}(s,a) = \underbrace{(\bar{r}_{t,h} - r_{t,h})(s,a)}_{\text{reward bias}} + \underbrace{(\widehat{r}_{t,h} - \bar{r}_{t,h})(s,a)}_{\text{reward variance}} +$$
$$\underbrace{[(\bar{P}_{t,h} - P_{t,h})V^\pi_{t,h+1}](s,a)}_{\text{transition bias}} + \underbrace{[(\widehat{P}_{t,h} - \bar{P}_{t,h})V_{t,h}](s,a)}_{\text{transition variance}} + \underbrace{[\bar{P}_{t,h}(V_{t,h+1} - V^\pi_{t,h+1})](s,a)}_{\text{difference in value functions of next step}}.$$

This differs from the error decomposition in the analysis of LSVI-UCB in several aspects: firstly, the variance terms are with respect the newly defined weighted MDP and not the true MPD. Secondly, we have additional reward and transition bias terms. Finally, the difference in the difference in value-functions at step $h+1$ hides also another bias term. Therefore, we need to carefully propagate bias terms through iteration.

The reward and transition bias terms are controlled by Lemma 9 using the fact that $\|V^\pi_{t,h}\|_\infty \leq H$. The difference in value-functions at step $h+1$ can be rewritten as $[P_{t,h}(V_{t,h+1} - V^\pi_{t,h+1})](s,a) + [(\bar{P}_{t,h} - P_{t,h})(V_{t,h+1} - V^\pi_{t,h+1})](s,a)$. We control the second term by applying again Lemma 9 since $\|V_{t,h+1} - V^\pi_{t,h+1}\|_\infty \leq H$.

It remains now the two variance terms. The reward variance is easy to control and it reduces simply to the bias due to the regularization as we assume

that r is a deterministic function. Note that the assumption of deterministic reward is not a limiting assumption since the contribution of a stochastic reward in the final regret has lower order term than the contribution of a stochastic transition. Controlling the transition variance is more involved. Basically, we would like use the concentration of weighted self-normalized processes [Russac et al., 2019] to get a high probability bound. However, as $V_{t,h+1}$ is estimated from past transitions and thus depends on the latter in a non-trivial way, we show a concentration bound that holds uniformly for all possible value functions generated by the algorithm. This done by using a union bound argument over an ϵ-net of the set of possible value functions with an appropriate value of ϵ.

Lemma 10. *For any $\delta \in (0,1)$, with probability at least $1-\delta/2$, we have for all $(t,h) \in [K] \times [H]$,*

$$\left\| \sum_{\tau=1}^{t-1} \eta^{-\tau} \phi_{\tau,h} \epsilon_{\tau,h} \right\|_{\widetilde{\Sigma}_{t,h}^{-1}} \leq CdH \sqrt{\log\left(\frac{dH\beta}{\lambda(1-\eta)} \cdot \frac{2}{\delta} \right)}$$

where $C > 0$ is an absolute constant.

Let $\texttt{bias} \triangleq \texttt{bias}_r + \texttt{bias}_P$ the total non-stationarity bias of the MDP. By deriving an appropriate value of β (see Lemma 20) and an induction arguments, we establish the optimism of our value estimates.

Lemma 11 (Optimism)**.** *There exists an absolute value c such that $\beta = cdH\sqrt{\iota}$ where $\iota = \log\left(\frac{2dH}{(1-\eta)\delta}\right)$, $\lambda = 1$ and for all $(s,a,t,h) \in \mathcal{S} \times \mathcal{A} \times [K] \times [H]$, we have with probability at least $1-\delta/2$*

$$Q_{t,h}(s,a) + 2H \sum_{h'=h}^{H} \texttt{bias}(t,h) \geq Q_{t,h}^{\star}(s,a) \tag{6.7.2}$$

6.8 Related Work

RL with linear function approximation: Provable algorithms with linear function approximation have seen a growing research interest in the recent literature. Under the assumption of stationary linear MDP, Jin et al. [2020b] propose an optimistic version of LSVI (LSVI-UCB) that achieves a regret of $\widetilde{\mathcal{O}}(d^{3/2}H^2K^{1/2})$ where the exploration is induced by adding a UCB bonus to the estimate of the action-value function. Whereas Zanette et al. [2020a] introduce a randomized version of LVSI that achieves $\widetilde{\mathcal{O}}(d^2H^2K^{1/2})$ regret where the exploration is induced by perturbing the estimate of the action-value function. Lately, Zanette et al. [2020b] consider a more general assumption, zero inherent Bellman error,

which states that the space of linear functions is close with respect to the Bellman operator (Note that linear MDPs have zero inherent Bellman error). Instead of adding UCB bonuses for every experienced states at each step $h \in [H]$, they propose to solve a global planning optimization program that returns an optimistic solution at the initial state, achieving $\widetilde{\mathcal{O}}(dH^2K^{1/2})$ regret. Yang and Wang [2019] study a slightly different assumption where the transition kernel admits a three-factor low-rank factorization $P(\cdot \mid \cdot) = \boldsymbol{\phi}(\cdot)^\top M^\star \boldsymbol{\psi}(\cdot)$. They propose a model-based algorithm that tries to learn the *core matrix* $M^\star$ and they show that it achieves $\widetilde{\mathcal{O}}(dH^2K^{1/2})$ regret.

Linear function approximations have also been studied in adversarial settings, where the reward function is allowed to change between episodes in an adversarial manner but the transition kernel stays the same. In the full-information setting, Cai et al. [2019] propose an optimistic policy optimization algorithm that achieves $\widetilde{\mathcal{O}}(dH^2K^{1/2})$ providing that the transition kernel has a linear structure $P_h(s' \mid s, a) = \boldsymbol{\psi}(s, a, s')^\top \theta_h$. In the bandit feedback setting, Neu and Olkhovskaya [2020] propose a new algorithm based on adversarial linear bandit that achieves $\widetilde{\mathcal{O}}((d|\mathcal{A}|)^{1/3}H^2K^{2/3})$ regret under the assumption that all action-value functions can be represented as linear functions.

Concurrently to our work, Zhou et al. [2020] also study non-stationary linear MDPs. They establish a lower bound of $\Omega(d^{2/3}H^2\Delta^{1/3}K^{2/3})$ and they propose a restart strategy that achieves the same dynamic regret as in our Corollary 1. Their algorithm consists in restarting periodically LSVI-UCB, and is thus markedly different from our approach. By throwing away historical data from time to time such a restart strategy would be best suited for abrupt changes in the environment, whereas our approach, by smoothly forgetting the past, would be more beneficial for gradually changing environments. Empirical comparison of both strategies in the bandit setting [Zhao et al., 2020] confirm this.

Non-stationary RL: Provably efficient algorithms for non-stationary RL in the tabular case have been introduced in several recent works. While Gajane et al. [2018] and Cheung et al. [2020] use a sliding-window approach, Ortner et al. [2019] implement a restart strategy where at each restart, past observations are discarded and new estimators for the reward and the transition kernel are built from scratch. Very recently, Domingues et al. [2020a] tackle non-stationary RL in continuous environments, where rewards and transition kernel are assumed to be Lipschitz with respect to some similarity metric over the state-action space. They propose a kernel-based algorithm with regret guarantee using time and space dependant smoothing kernels.

6.9 Conclusion

In this paper, we studied the problem of RL with linear function approximation in a changing environment where the reward and the transition kernel can change from time to time as long as the total changes are bounded by some variation budget. We introduced a provably efficient algorithm in this setting. The algorithm uses a discount factor to reduce the influence of the past and estimates the Q-value's parameters through weighted LSVI. We revisited as well the linear bandit setting. We pointed out a serious technical problem in the analysis of all forgetting strategies. Then, we provide a new regret analysis of these algorithms.

Limitations: In order to obtain theoretical guarantees, we need to make some assumptions such as Linear MDPs. Assumptions weaker than linear MDP either result in computationally inefficient algorithms (as in Zanette et al. [2020b]) or require the transition to be deterministic [Du et al., 2020]. Furthermore, our $\widetilde{\mathcal{O}}(T^{3/4})$ regrets for both bandits and MDPs don't match the $\Omega(T^{2/3})$ lower bounds for these problems. Forgetting strategies have been mistakenly believed optimal in linear bandits. In contrast, our work shows that the latter is not true and leaves the question of minimax rate open again. It is an interesting direction to explore in future work.

7 Learning One Representation to Optimize All Rewards

7.1 Prologue to the Contribution

7.1.1 Article Details

This chapter is based on the article *«Learning One Representation to Optimize All Rewards»* [Touati and Ollivier, 2021], joint work with Yann Ollivier. This paper was presented as long oral in the self-supervision for reinforcement learning workshop at ICLR 2021. It was also accepted at NeurIPS 2021. I am the first author. Yann Ollivier was at the origin of the idea of the project and had already developed a large part of the theory. I was in charge of implementing the new algorithm and performing the experiments. I also contributed to further develop the theoretical analysis.

7.1.2 Context

We have focused so far on the classical RL paradigm which aims to maximize the cumulative reward when the agent has access to the reward signal. Despite being able to capture many AI applications, this reward-driven framework leads to a task-specific agent that is only able to solve the task at hand and needs to be trained from the scratch to adapt to new tasks. It is obvious here that we overlook a lot of information about the environment that could be leveraged across tasks. In this project, we are motivated by the following research problem:

Imagine you have a fixed environment where you can perform actions but you receive no reward information. Your mission is to build a summary of the environment using these reward-free interactions, such that as soon as I describe a reward function, you immediately know what to do to maximize your reward.

This is a step towards building controllable agents able to follow instructions.

7.1.3 Paper Abstract

We introduce the *forward-backward* (FB) representation of the dynamics of a reward-free Markov decision process. It provides explicit near-optimal policies for any reward specified a posteriori. During an unsupervised phase, we use

reward-free interactions with the environment to learn two representations via off-the-shelf deep learning methods and temporal difference (TD) learning. In the test phase, a reward representation is estimated either from reward observations or an explicit reward description (e.g., a target state). The optimal policy for that reward is directly obtained from these representations, with no planning. We assume access to an exploration scheme or replay buffer for the first phase.

The unsupervised FB loss is well-principled: if training is perfect, the policies obtained are provably optimal for any reward function. With imperfect training, the sub-optimality is proportional to the unsupervised approximation error. The FB representation learns long-range relationships between states and actions, via a predictive occupancy map, without having to synthesize states as in model-based approaches.

Our approach compares well to goal-oriented RL algorithms on discrete and continuous mazes, pixel-based MsPacman, and the FetchReach virtual robot arm. We also illustrate how the agent can immediately adapt to new tasks beyond goal-oriented RL.

Our Code is available at: https://github.com/ahmed-touati/controllable_agent.

7.2 INTRODUCTION

We consider one kind of unsupervised reinforcement learning problem: Given a Markov decision process (MDP) but no reward information, is it possible to learn and store a compact object that, for any reward function specified later, provides the optimal policy for that reward, with a minimal amount of additional computation? In a sense, such an object would encode in a compact form the solutions of all possible planning problems in the environment. This is a step towards building agents that are fully controllable after first exploring their environment in an unsupervised way.

Goal-oriented RL methods [Andrychowicz et al., 2017, Plappert et al., 2018] compute policies for a series of rewards specified in advance (such as reaching a set of target states), but cannot adapt in real time to new rewards, such as weighted combinations of target states or dense rewards.

Learning a model of the world is another possibility, but it still requires explicit planning for each new reward; moreover, synthesizing accurate trajectories of states over long time ranges has proven difficult [Talvitie, 2017, Ke et al., 2018].

Instead, we exhibit an object that is both simpler to learn than a model of the world, and contains the information to recover near-optimal policies for any reward provided a posteriori, without a planning phase.

Borsa et al. [2018] learn optimal policies for all rewards that are linear combinations of a finite number of feature functions provided in advance by the user. This limits applications: e.g., goal-oriented tasks would require one feature per goal state, thus using infinitely many features in continuous spaces. We reuse a policy parameterization from Borsa et al. [2018], but introduce a novel representation with better properties, based on state occupancy prediction instead of expected featurizations. We use theoretical advances on successor state learning from Blier et al. [2021]. We obtain the following.

- We prove the existence of a learnable "summary" of a reward-free discrete or continuous MDP, that provides an explicit formula for optimal policies for any reward specified later. This takes the form of a pair of representations $F\colon S \times A \times Z \to Z$ and $B\colon S \times A \to Z$ from state-actions into a representation space $Z \simeq \mathbb{R}^d$, with policies $\pi_z(s) \triangleq \operatorname{argmax}_a F(s,a,z)^\top z$. Once a reward is specified, a value of z is computed from reward values and B; then π_z is used. Rewards may be specified either explicitly as a function, or as target states, or by samples as in usual RL setups.

- We provide a well-principled unsupervised loss for F and B. If FB training is perfect, then the policies are provably optimal for all rewards (Theorem 1). With imperfect training, sub-optimality is proportional to the FB training error (Theorems 5–6). In finite spaces, perfect training is possible with large enough dimension d (Proposition 6).

  Explicitly, F and B are trained so that $F(s,a,z)^\top B(s',a')$ approximates the long-term probability to reach s' from s if following π_z. This is akin to a model of the environment, without synthesizing state trajectories.

- We provide a TD-like algorithm to train F and B for this unsupervised loss, with function approximation, adapted from recent methods for successor states [Blier et al., 2021]. No sparse rewards are used: every transition reaches some state s', so every step is exploited. As usual with TD, learning seeks a fixed point but the loss itself is not observable.

- We prove viability of the method on several environments from mazes to pixel-based MsPacman and a virtual robotic arm. For single-state rewards (learning to reach arbitrary states), we provide quantitative comparisons with goal-oriented methods such as HER. (Our method is not a substitute for HER: in principle they could be combined, with HER improving replay buffer management for our method.) For more general rewards, which cannot be tackled a posteriori by trained goal-oriented models, we provide qualitative examples.

- We also illustrate qualitatively the sub-optimalities (long-range behavior

is preserved but local blurring of rewards occurs) and the representations learned.

7.3 Problem and Notation

Let $\mathcal{M} = (S, A, P, \gamma)$ be a reward-free Markov decision process.

For any policy $\pi :: S \to \mathrm{Prob}(A)$ and state-action (s_0, a_0), define the *successor measure* $M^\pi(s_0, a_0, \cdot)$ as the measure over $S \times A$ representing the expected discounted time spent in each set $X \subset S \times A$:

$$M^\pi(s_0, a_0, X) \triangleq \sum_{t \geq 0} \gamma^t \Pr\left((s_t, a_t) \in X \mid s_0,\, a_0,\, \pi\right) \tag{7.3.1}$$

for each $X \subset S \times A$. Viewing M as a measure deals with both discrete and continuous spaces.

We consider the following informal problem: Given a reward-free MDP (S, A, P, γ), can we compute a convenient learnable object E such that, once a reward function $r \colon S \times A \to \mathbb{R}$ is specified, we can easily (with no planning) compute, from E and r, a policy π whose performance is close to maximal?

7.4 Encoding All Optimal Policies via the Forward-Backward Representation

We first present forward-backward (FB) representations of a reward-free MDP as a way to summarize all optimal policies via explicit formulas. The resulting learning procedure is described in Section 7.5.

Core idea. The main algebraic idea is as follows. Assume, at first, that S is finite. For a fixed policy, the Q-function depends lineary on the reward: namely, $Q_r^\pi(s, a) = \sum_{s', a'} M^\pi(s, a, s', a') r(s', a')$ where $M^\pi(s, a, s', a') = \sum_{t \geq 0} \gamma^t \Pr\left((s_t, a_t) = (s', a') | s, a, \pi\right)$. This rewrites as $Q_r^\pi = M^\pi r$ viewing everything as vectors and matrices indexed by state-actions.

Now let $(\pi_z)_{z \in \mathbb{R}^d}$ be any family of policies parameterized by z. Assume that for each z, we can find $d \times (S \times A)$-matrices F_z and B such that $M^{\pi_z} = F_z^\top B$. Then $Q_r^{\pi_z} = F_z^\top B r$. Specializing to $z_R \triangleq Br$, the Q-function of policy π_{z_R} on reward r is $Q_r^{\pi_{z_R}} = F_{z_R}^\top z_R$. So far π_z was unspecified; but if we define $\pi_z(s) \triangleq \mathrm{argmax}_a (F_z^\top z)_{sa}$ at each state s, then by definition, π_{z_R} is the greedy policy with respect to $F_{z_R}^\top z_R$. At the same time, $F_{z_R}^\top z_R$ is the Q-function of π_{z_R} for reward r:

thus, π_{z_R} is the greedy policy of its own Q-function, and is therefore optimal for reward r.

Thus, if we manage to find F, B, and π_z such that $\pi_z = \operatorname{argmax} F_z^\top z$ and $F_z^\top B = M^{\pi_z}$ for all $z \in \mathbb{R}^d$, then we obtain the optimal policy for any reward r, just by computing Br and applying policy π_{Br}.

This criterion on (F, B, π_z) is entirely unsupervised. Since F and B depend on π_z but π_z is defined via F, this is a fixed point equation. An exact solution exists for d large enough (Appendix, Prop. 6), while a smaller d provides lower-rank approximations $M^{\pi_z} \approx F_z^\top B$. In Section 7.5 we present a well-grounded algorithm to learn such F, B, and π_z.

In short, we learn two representations F and B such that $F(s_0, a_0, z)^\top B(s', a')$ is approximately the long-term probability $M^{\pi_z}(s_0, a_0, s', a')$ to reach (s', a') if starting at (s_0, a_0) and following policy π_z. Then all optimal policies can be computed from F and B. We think of F as a representation of the future of a state, and B as the ways to reach a state (Appendix 7.8.4): if $F^\top B$ is large, then the second state is reachable from the first. This is akin to a model of the environment, without synthesizing state trajectories.

General statement. In continuous spaces with function approximation, F_z and B become functions $S \times A \to \mathbb{R}^d$ instead of matrices; since F_z depends on z, F itself is a function $S \times A \times \mathbb{R}^d \to \mathbb{R}^d$. The sums over states will be replaced with expectations under the data distribution ρ.

Definition 4 (Forward-backward representation)**.** *Let $Z = \mathbb{R}^d$ be a representation space, and let ρ be a measure on $S \times A$. A pair of functions $F\colon S \times A \times Z \to Z$ and $B\colon S \times A \to Z$, together with a parametric family of policies $(\pi_z)_{z \in Z}$, is called a* forward-backward representation *of the MDP with respect to ρ, if the following conditions hold for any $z \in Z$ and $(s, a), (s_0, a_0) \in S \times A$:*

$$\pi_z(s) = \operatorname*{argmax}_a F(s, a, z)^\top z, \qquad M^{\pi_z}(s_0, a_0, \mathrm{d}s, \mathrm{d}a) = F(s_0, a_0, z)^\top B(s, a)\rho(\mathrm{d}s, \mathrm{d}a) \tag{7.4.1}$$

where M^π is the successor measure defined in (7.3.1), and the last equality is between measures.

Theorem 1 (FB representations encode all optimal policies)**.** *Let $(F, B, (\pi_z))$ be a forward-backward representation of a reward-free MDP with respect to some measure ρ.*

Then, for any bounded reward function $r\colon S \times A \to \mathbb{R}$, the following holds. Set

$$z_R \triangleq \int_{s,a} r(s, a) B(s, a)\, \rho(\mathrm{d}s, \mathrm{d}a). \tag{7.4.2}$$

assuming the integral exists. Then π_{z_R} is an optimal policy for reward r in the MDP. Moreover, the optimal Q-function $Q^\star$ for reward r is $Q^\star(s, a) = F(s, a, z_R)^\top z_R$.

For instance, for a single reward located at state-action (s, a), the optimal policy is π_{z_R} with $z_R = B(s, a)$. (In that case the factor $\rho(\mathrm{d}s, \mathrm{d}a)$ does not matter because scaling the reward does not change the optimal policy.)

We present in Section 7.5 an algorithm to learn FB representations. The measure ρ will be the distribution of state-actions visited in a training set or under an exploration policy: then $z_R = \mathbb{E}_{(s,a)\sim\rho}[r(s, a)B(s, a)]$ can be obtained by sampling from visited states.

In finite spaces, exact FB representations exist, provided the dimension d is larger than $\#S \times \#A$ (Appendix, Prop. 6). In infinite spaces, arbitrarily good approximations can be obtained by increasing d, corresponding to a rank-d approximation of the cumulated transition probabilities M^π. Importantly, the optimality guarantee extends to approximate F and B, with optimality gap proportional to $\left\|F^\top B - M^{\pi_z}\right\|/\rho$ (Appendix, Theorems 5–6 with various norms on $\left\|F^\top B - M^\pi\right\|/\rho$). For instance, if, for some reward r, the error $\left|F(s_0, a_0, z_R)^\top B(s, a) - M^{\pi_{z_R}}(s_0, a_0, \mathrm{d}s, \mathrm{d}a)/\rho(\mathrm{d}s, \mathrm{d}a)\right|$ is at most ϵ on average over $(s, a) \sim \rho$ for every (s_0, a_0), then π_{z_R} is $3\epsilon \left\|r\right\|_\infty / (1-\gamma)$-optimal for r.

These results justify using some norm over $\left|F^\top B - M^{\pi_z}/\rho\right|$, averaged over $z \in \mathbb{R}^d$, as a training loss for unsupervised reinforcement learning. (Below, we average over $z \in \mathbb{R}^d$ from a fixed rescaled Gaussian. If prior information is available on the rewards r, the corresponding distribution of z_R may be used instead.)

If B is fixed in advance and only F is learned, the method has similar properties to successor features based on B (Appendix 7.8.4). But one may set a large d and let B be learned: arguably, by Theorem 1, the resulting features "linearize" optimal policies as much as possible. The features learned in F and B may have broader interest.

7.5 Learning and Using Forward-Backward Representations

Our algorithm starts with an *unsupervised learning phase*, where we learn the representations F and B in a reward-free way, by observing state transitions in the environment, generated from any exploration scheme. Then, in a *reward estimation phase*, we estimate a policy parameter $z_R = \mathbb{E}[r(s, a)B(s, a)]$ from some reward observations, or directly set z_R if the reward is known (e.g., set $z_R = B(s, a)$ to reach a known target (s, a)). In the *exploitation phase*, we directly use the policy $\pi_{z_R}(s) = \mathrm{argmax}_a F(s, a, z_R)^\top z_R$.

The unsupervised learning phase. No rewards are used in this phase, and no family of tasks has to be specified manually. F and B are trained off-policy from

observed transitions in the environment, to approximate the successor density: $F^\top(s_0, a_0, z)B(s', a') \approx m^{\pi_z}(s_0, a_0, s', a')$ for every z. Training is based on the Bellman equation for the successor measure M^π,

$$M^\pi(s_0, a_0, \{(s', a')\}) = \mathbb{1}_{s_0=s', a_0=a'} + \gamma \mathbb{E}_{s_1 \sim P(\mathrm{d}s_1|s_0,a_0)} M^\pi(s_1, \pi(s_1), \{(s', a')\}). \tag{7.5.1}$$

We leverage a well-principled algorithm from Blier et al. [2021] in the single-policy setting: it learns the successor density m^π of a policy π without using the sparse reward $\mathbb{1}_{s_0=s', a_0=a'}$ (which would vanish in continuous spaces). This algorithm uses a parametric model $m^\pi_\theta(s_0, a_0, s', a')$. Given an observed transition (s_0, a_0, s_1) from the training set, generate an action $a_1 \sim \pi(a_1|s_1)$, and sample another state-action (s', a') from the training set, independently from (s_0, a_0, s_1). Then update the parameter θ by $\theta \leftarrow \theta + \eta\delta\theta$ with learning rate η and

$$\delta\theta \triangleq \partial_\theta m^\pi_\theta(s_0, a_0, s_0, a_0) + \partial_\theta m^\pi_\theta(s_0, a_0, s', a') \times \left(\gamma\, m^\pi_\theta(s_1, a_1, s', a') - m^\pi_\theta(s_0, a_0, s', a')\right) \tag{7.5.2}$$

This computes the density m^π of M^π with respect to the distribution ρ of state-actions in the training set. Namely, the true successor state density m^π is a fixed point of (7.5.2) in expectation Blier et al. [2021]. Variants exist, such as using a target network for $m^\pi_\theta(s_1, a_1, s', a')$ on the right-hand side, as in DQN.

Thus, we first choose a parametric model F_θ, B_θ for the representations F and B, and set $m^{\pi_z}_\theta(s_0, a_0, s', a') \triangleq F_\theta(s_0, a_0, z)^\top B_\theta(s', a')$. Then we iterate the update (7.5.2) over many state-actions and values of z. This results in Algorithm 7. At each step, a value of z is picked at random, together with a batch of transitions (s_0, a_0, s_1) and a batch of state-actions (s', a') from the training set, with (s', a') independent from z and (s_0, a_0, s_1).

For sampling z, we use a fixed distribution (rescaled Gaussians, see Appendix E.2). Any number of values of z may be sampled: this does not use up training samples. We use a target network with soft updates (Polyak averaging) as in DDPG. For training we also replace the greedy policy $\pi_z = \mathrm{argmax}_a F(s, a, z)^\top z$ with a regularized version $\pi_z = \mathrm{softmax}(F(s, a, z)^\top z/\tau)$ with fixed temperature τ (Appendix E.2). Since there is unidentifiability between F and B (Remark 4), we normalize B via an auxiliary loss in Algorithm 7.

For exploration in this phase, we use the policies being learned: the exploration policy chooses a random value of z from some distribution (e.g., Gaussian), and follows π_z for some time (Algorithm 7). However, the algorithm can also work from an existing dataset of off-policy transitions.

The reward estimation phase. Once rewards are available, we estimate a reward representation (policy parameter) z_R by weighing the representation B by the reward:

$$z_R \triangleq \mathbb{E}[r(s, a)B(s, a)] \tag{7.5.3}$$

where the expectation must be computed over the same distribution ρ of state-actions (s, a) used to learn F and B (see Section 7.8.5 for using a different distribution). Thus, if the reward is black-box as in standard RL algorithms, then the exploration policy has to be run again for some time, and z_R is obtained by averaging $r(s, a)B(s, a)$ over the states visited.

If the reward is known explicitly, this phase is unnecessary. For instance, if the reward is to reach a target state-action (s_0, a_0) while avoiding some forbidden state-actions $(s_1, a_1), ..., (s_k, a_k)$, one may directly set

$$z_R = B(s_0, a_0) - \lambda \sum B(s_i, a_i) \tag{7.5.4}$$

where the constant λ adjusts the negative reward for visiting a forbidden state. This can be used for goal-oriented RL.

If the reward is known algebraically as a function $r(s, a)$, then z_R may be computed by averaging the function $r(s, a)B(s, a)$ over a replay buffer from the unsupervised training phase. We may also use a reward model $\hat{r}(s, a)$ of $r(s, a)$ trained on some reward observations from any source. An approximate value for z_R still provides an approximately optimal policy (Proposition 7 and Theorem. 7).

The exploitation phase. Once the reward representation z_R has been estimated, the Q-function is estimated as

$$Q(s, a) = F(s, a, z_R)^\top z_R. \tag{7.5.5}$$

The corresponding policy $\pi_{z_R}(s) = \text{argmax}_a\, Q(s, a)$ is used for exploitation.

Fine-tuning was not needed in our experiments, but it is possible to fine-tune the Q-function using actual rewards, by setting $Q(s, a) = F(s, a, z_R)^\top z_R + q_\theta(s, a)$ where the fine-tuning model q_θ is initialized to 0 and learned via any standard Q-learning method.

Incorporating prior information on rewards in B. Trying to plan in advance for all possible rewards in an arbitrary environment may be too generic and problem-agnostic, and become difficult in large environments, requiring long exploration and a large d to accommodate all rewards. In practice, we are often interested in rewards depending, not on the full state, but only on a part or some features of the state (e.g., a few components of the state, such as the position of an agent, or its neighbordhood, rather than the full environment).

If this is known in advance, the representation B can be trained on that part of the state only, with the same theoretical guarantees (Appendix, Theorem 2). F still needs to use the full state as input. This way, the FB model of the transition probabilities (7.3.1) only has to learn the future probabilities of the part of interest in (s', a'), based on the full initial state (s_0, a_0). Explicitly, if $\phi\colon S \times A \to G$

Algorithm 7 FB algorithm: Unsupervised Phase

Inputs: replay buffer $\mathcal{D}$, Polyak coefficient α , ν a probability distribution over $\mathbb{R}^d$, randomly initialized networks F_θ and B_ω, learning rate η, mini-batch size b, number of episodes E, number of gradient updates N, temperature τ and regularization coefficient λ.
for $m = 1, \dots$ **do**
  /* Collect E episodes
  for episode $e = 1, \dots E$ **do**
    Sample $z \sim \nu$
    Observe an initial state s_0
    for $t = 1, \dots$ **do**
      Select an action a_t according to some behaviour policy (e.g the ϵ-greedy with respect to $F_\theta(s_t, a, z)^\top z$)
      Observe next state s_{t+1}
      Store transition (s_t, a_t, s_{t+1}) in the replay buffer $\mathcal{D}$
    end for
  end for
  /* Perform N stochastic gradient descent updates
  for $n = 1 \dots N$ **do**
    Sample a mini-batch of transitions $\{(s_i, a_i, s_{i+1})\}_{i \in I} \subset \mathcal{D}$ of size $|I| = b$.
    Sample a mini-batch of target state-action pairs $\{(s'_i, a'_i)\}_{i \in I} \subset \mathcal{D}$ of size $|I| = b$.
    Sample a mini-batch of $\{z_i\}_{i \in I} \sim \nu$ of size $|I| = b$.
    Set $\pi_{z_i}(\cdot \mid s_{i+1}) = \texttt{softmax}(F_{\theta^-}(s_{i+1}, \cdot, z_i)^\top z_i / \tau)$
    $\mathscr{L}(\theta, \omega) = -\frac{1}{b} \sum_{i \in I} F_\theta(s_i, a_i, z_i)^\top B_\omega(s_i, a_i) +$
    $\frac{1}{2b^2} \sum_{i,j \in I^2} \left(F_\theta(s_i, a_i, z_i)^\top B_\omega(s'_j, a'_j) - \gamma \sum_{a \in A} \pi_{z_i}(a \mid s_{i+1}) \cdot F_{\theta^-}(s_{i+1}, a, z_i)^\top B_{\omega^-}(s'_j, a'_j) \right)^2$
    /* Compute orthonormality regularization loss
    $\mathscr{L}_{\texttt{reg}}(\omega) = -\frac{1}{b} \sum_{i \in I} B_\omega(s_i, a_i)^\top \texttt{stop-gradient}(B_\omega(s_i, a_i)) + \frac{1}{b^2} \sum_{i,j \in I^2} B_\omega(s_i, a_i)^\top \texttt{stop-gradient}(B_\omega(s'_j, a'_j)) \cdot \texttt{stop-gradient}(B_\omega(s_i, a_i)^\top B_\omega(s'_j, a'_j))$
    Update $\theta \leftarrow \theta - \eta \nabla_\theta \mathscr{L}(\theta, \omega)$ and $\omega \leftarrow \omega - \eta \nabla_\omega (\mathscr{L}(\theta, \omega) + \lambda \cdot \mathscr{L}_{\texttt{reg}}(\omega))$
  end for
  /* Update target network parameters
  $\theta^- \leftarrow \alpha \theta^- + (1 - \alpha)\theta$
  $\omega^- \leftarrow \alpha \omega^- + (1 - \alpha)\omega$
end for

is a feature map to some features $g = \phi(s, a)$, and if we know that the reward will be a function $R(g)$, then Theorem 1 still holds with $B(g)$ everywhere instead

of $B(s,a)$, and with the successor density $m^\pi(s_0,a_0,g)$ instead of $m^\pi(s_0,a_0,s',a')$ (Theorem 2). Rewards can be arbitrary functions of g, so this is more general than Borsa et al. [2018] which only considers rewards linear in g. For instance, in MsPacman below, we let g be the 2D position (x,y) of the agent, so we can optimize any reward function that depends on this position.

Limitations. First, this method does not solve exploration: it assumes access to a good exploration strategy. (Here we used the learned π_z with random values of z, corresponding to random rewards.)

Next, this task-agnostic approach is relevant if the reward is not known in advance, but may not bring the best performance on a particular reward. Mitigation strategies include: increasing d; using prior information on rewards by including relevant variables into B, as discussed above; and fine-tuning the Q-function at test time based on the initial $F^\top B$ estimate.

Indeed, as reward functions are represented by a d-dimensional vector $z_R = \mathbb{E}[r.B]$, some information about the reward is necessarily lost. Any reward uncorrelated to B is treated as 0. The necessary dimension d for good behavior may be large. Still, $d \approx 100$ worked in our experiments, and Section 7.8.2 argues theoretically that $d = O(n)$ is enough for navigation on an n-dimensional grid.

We expect this method to have an implicit bias for long-range behavior (spatially smooth rewards), while local details of the reward function may be blurred. Indeed, $F^\top B$ is optimized to approximate the successor measure $M^\pi = \sum_t \gamma^t (P^\pi)^t$ with $(P^\pi)^t$ the t-step transition kernel for each policy π. The rank-d approximation will favor large eigenvectors of P^π, i.e., small eigenvectors of the Markov chain Laplacian $\mathrm{Id} - \gamma P^\pi$. These loosely correspond to long-range (low-frequency) behavior Mahadevan and Maggioni [2007]: presumably, F and B will learn spatially smooth rewards first. Indeed, experimentally, a small d leads to spatial blurring of rewards and Q-functions (Fig. 7.3). Arguably, without any prior information this is a reasonable prior; Stachenfeld et al. [2017] have argued for the cognitive relevance of low-dimensional approximations of successor representations.

Variance is a potential issue in larger environments, although this did not arise in our experiments. Learning m^π requires sampling a state-action (s_0,a_0) and an independent state-action (s',a'). In large spaces, most state-action pairs will be unrelated. A possible mitigation is to use strategies such as Hindsight Experience Replay Andrychowicz et al. [2017] to select goals related to the current state-action. The following may help a lot: the update of F and B decouples as an expectation over (s_0,a_0), times an expectation over (s',a'). Thus, by estimating these expectations by a moving average over a dataset, it is easy to have many pairs (s_0,a_0) interact with many (s',a'). The cost is handling full $d \times d$ matrices. This will be explored in future work.

7.6 EXPERIMENTS

We first consider the task of reaching arbitrary goal states. For this, we can make quantitative comparisons to existing goal-oriented baselines. Next, we illustrate qualitatively some tasks that cannot be tackled a posteriori by goal-oriented methods, such as introducing forbidden states. Finally, we illustrate some of the representations learned.

7.6.1 Environments and Experimental Setup

We run our experiments on a selection of environments that are diverse in term of state space dimensionality, stochasticity and dynamics.

- Discrete Maze is the classical gridworld with four rooms. States are represented by one-hot unit vectors.
- Continuous Maze is a two dimensional environment with impassable walls. States are represented by their Cartesian coordinates $(x, y) \in [0, 1]^2$. The execution of one of the actions moves the agent in the desired direction, but with normal random noise added to the position of the agent.
- FetchReach is a variant of the simulated robotic arm environment from Plappert et al. [2018] using discrete actions instead of continuous actions. States are 10-dimensional vectors consisting of positions and velocities of robot joints.
- Ms. Pacman is a variant of the Atari 2600 game Ms. Pacman, where an episode ends when the agent is captured by a monster Rauber et al. [2018]. States are obtained by processing the raw visual input directly from the screen. Frames are preprocessed by cropping, conversion to grayscale and downsampling to 84×84 pixels. A state s_t is the concatenation of $(x_{t-12}, x_{t-8}, x_{t-4}, x_t)$ frames, i.e. an $84 \times 84 \times 4$ tensor. An action repeat of 12 is used. As Ms. Pacman is not originally a multi-goal domain, we define the goals as the 148 reachable coordinates (x, y) on the screen; these can be reached only by learning to avoid monsters.

For all environments, we run algorithms for 800 epochs. Each epoch consists of 25 cycles where we interleave between gathering some amount of transitions, to add to the replay buffer, and performing 40 steps of stochastic gradient descent on the model parameters. To collect transitions, we generate episodes using some behavior policy. For both mazes, we use a uniform policy while for FetchReach and Ms. Pacman, we use an ϵ-greedy policy with respect to the current approximation $F(s, a, z)^\top z$ for a sampled z. At evaluation time, ϵ-greedy policies are also used, with a smaller ϵ. More details are given in Appendix E.2.

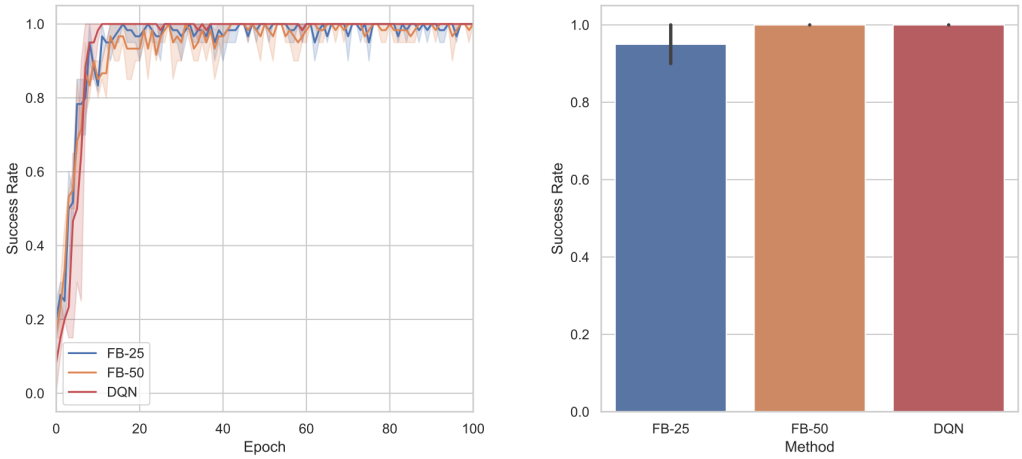

Figure 7.1: Comparative performance of FB for different dimensions and DQN in the FetchReach. **Left**: success rate averaged over 20 randomly selected goals as function of the first 100 training epochs. **Right**: success rate averaged over 20 random goals after 800 training epochs.

7.6.2 Goal-Oriented Setting: Quantitative Comparisons

We investigate the FB representation over goal-reaching tasks and compare it to goal-oriented baselines: DQN[1], and DQN with HER when needed. We define sparse reward functions. For Discrete Maze, the reward function is equal to one when the agent's state is equal exactly to the goal state. For Discrete Maze, we measured the quality of the obtained policy to be the ratio between the true expected discounted reward of the policy for its goal and the true optimal value function, on average over all states. For the other environments, the reward function is equal to one when the distance of the agent's position and the goal position is below some threshold, and zero otherwise. We assess policies by computing the average success rate, i.e the average number of times the agent successfully reaches its goal.

Figs. 7.1 and 7.2 show the comparative performance of FB for different dimensions d, and DQN respectively in FetchReach and Ms. Pacman (similar results in Discrete and Continuous Mazes are provided in Appendix E.2). In Ms. Pacman, DQN totally fails to learn and we had to add HER to make it work. The performance of FB consistently increases with the dimension d and the best dimension matches the performance of the goal-oriented baseline.

In Discrete Maze, we observe a drop of performance for $d = 25$ (Appendix E.2, Fig. E.1): this is due to the spatial smoothing induced by the small

[1]Here DQN is short for goal-oriented DQN, $Q(s, a, g)$.

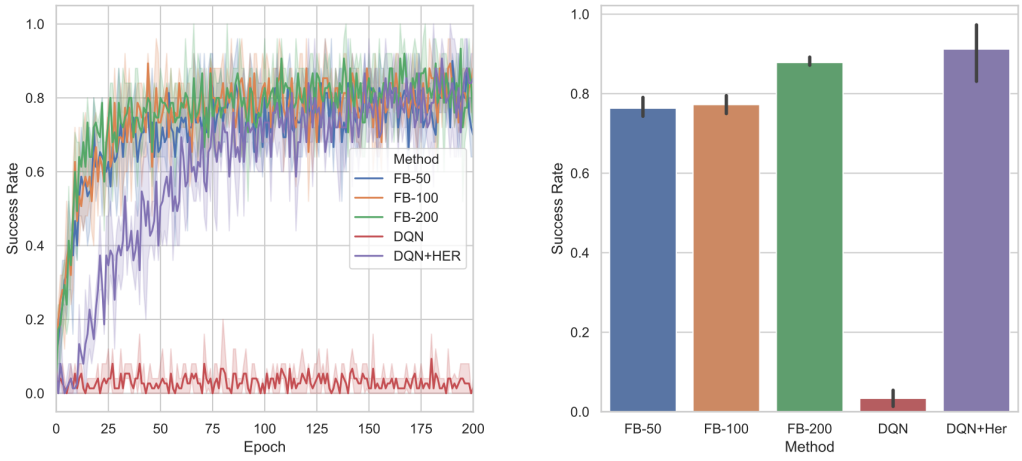

Figure 7.2: Comparative performance of FB for different dimensions and DQN in Ms. Pacman. **Left**: success rate averaged over 20 randomly selected goals as function of the first 200 training epochs. **Right**: success rate averaged over the goal space after 800 training epochs.

rank approximation and the reward being nonzero only if the agent is exactly at the goal. This spatial blurring is clear on heatmaps for $d = 25$ vs $d = 75$ (Fig. 7.3). With $d = 25$ the agent often stops right next to its goal.

To evaluate the sample efficiency of FB, after each epoch, we evaluate the agent on 20 randomly selected goals. Learning curves are reported in Figs. 7.1 and 7.2 (left). In all environments, we observe no loss in sample efficiency compared to the goal-oriented baseline. In Ms. Pacman, FB even learns faster than DQN+HER.

7.6.3 More Complex Rewards: Qualitative Results

We now investigate FB's ability to generalize to new tasks that cannot be solved by an already trained goal-oriented model: reaching a goal with forbidden states imposed a posteriori, reaching the nearest of two goals, and choosing between a small, close reward and a large, distant one.

First, for the task of reaching a target position g_0 ☆ while avoiding some forbidden positions $g_1, \dots g_k$ ●, we set $z_R = B(g_1) - \lambda \sum_{i=1}^{k} B(g_i)$ and run the corresponding ϵ-greedy policy defined by $F(s, a, z_R)^\top z_R$. Fig. 7.4 shows the resulting trajectories, which succeed at solving the task for the different domains. In Ms. Pacman, the path is suboptimal (though successful) due to the sudden appearance of a monster along the optimal path. (We only plot the initial frame; see the full series of frames along the trajectory in Appendix E.2, Fig. E.9.) Fig. 7.6

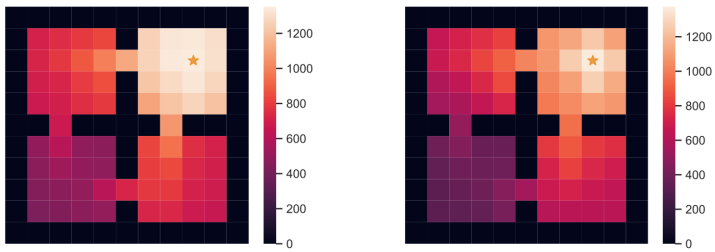

Figure 7.3: Heatmap of $\max_a F(s,a,z_R)^\top z_R$ for $z_R = B(\bigstar)$ **Left**: $d = 25$. **Right**: $d = 75$.

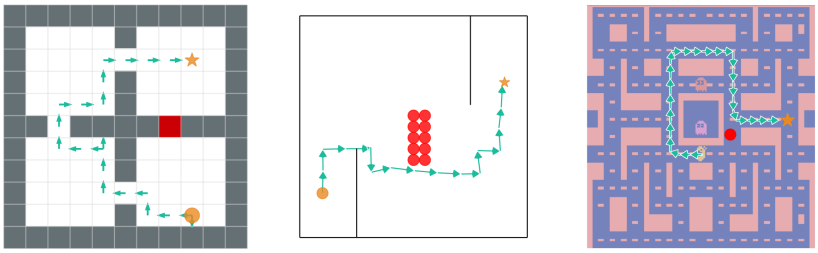

Figure 7.4: Trajectories generated by the $F^\top B$ policies for the task of reaching a target position (star shape $\bigstar$) while avoiding forbidden positions (red circle).

(left) provides a contour plot of $\max_{a\in A} F(s,a,z_R)^\top z_R$ for the continuous maze and shows the landscape shape around the forbidden regions.

Next, we consider the task of reaching the closest target among two equally rewarding positions g_0 and g_1, by setting $z_R = B(g_0) + B(g_1)$. The optimal Q-function is *not* a linear combination of the Q-functions for g_0 and g_1. Fig. 7.5 shows successful trajectories generated by the policy π_{z_R}. On the contour plot of $\max_{a\in A} F(s,a,z_R)^\top z_R$ in Fig. 7.6 (right), the two rewarding positions appear as basins of attraction. Similar results for a third task are shown in Appendix E.2: introducing a "distracting" small reward next to the initial position of the agent, with a larger reward further away.

7.6.4 Embedding Visualizations

We visualize the learned FB state embeddings for Continuous Maze by projecting them into 2-dimensional space using t-SNE [Van der Maaten and Hinton, 2008] in Fig. 7.7. For the forward embeddings, we set $z = 0$ corresponding to the uniform policy. We can see that FB partitions states according to the topology induced by the dynamics: states on opposite sides of walls are separated in the representation space and states on the same side lie together. Appendix E.2 in-

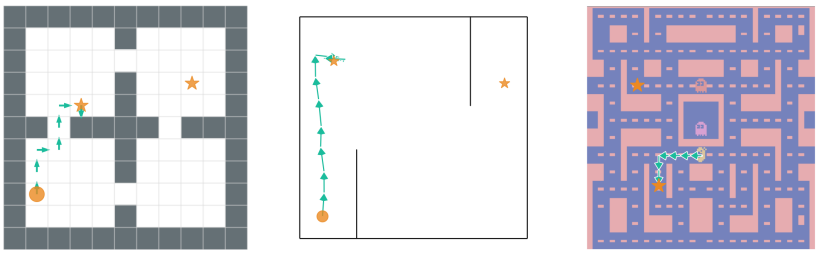

Figure 7.5: Trajectories generated by the $F^\top B$ policies for the task of reaching the closest among two equally rewarding positions (star shapes ★). (Optimal Q-values are not linear over such mixtures.)

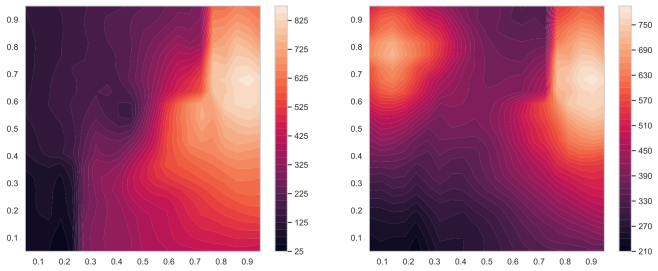

Figure 7.6: Contour plot of $\max_{a \in A} F(s, a, z_R)^\top z_R$ in Continuous Maze. **Left**: for the task of reaching a target while avoiding a forbidden region, **Right**: for two equally rewarding targets.

cludes embedding visualizations for different z and for Discrete Maze and Ms. Pacman.

7.7 Related work

Borsa et al. [2018] learn optimal policies for rewards that are linear combinations of a finite number of feature functions provided in advance by the user. This approach cannot tackle generic rewards or goal-oriented RL: this would require introducing one feature per possible goal state, requiring infinitely many features in continuous spaces.

Our approach does not require user-provided features describing the future tasks, thanks to using successor *states* [Blier et al., 2021] where Borsa et al. [2018] use successor *features*. Schematically, and omitting actions, successor features start with user-provided features ϕ, then learn ψ such that $\psi(s_0) = \sum_{t \geq 0} \gamma^t \mathbb{E}[\phi(s_t) \mid s_0]$. This limits applicability to rewards that are linear combinations of ϕ. Here we use successor *state* probabilities, namely, we learn two

Figure 7.7: Visualization of FB embedding vectors on Continuous Maze after projecting them in two-dimensional space with t-SNE. **Left**: the states to be mapped. **Middle**: the F embedding. **Right**: the B embedding. The walls appear as large dents; the smaller dents correspond to the number of steps needed to get past a wall.

representations F and B such that $F(s_0)^\top B(s') = \sum_{t\geq 0} \gamma^t \Pr(s_t = s' \mid s_0)$. This does not require any user-provided input.

We use a similar parameterization of policies by $F(s, a, z)^\top z$ as in Borsa et al. [2018], for similar reasons, although z encodes a different object.

Successor representations where first defined in Dayan [1993] for finite spaces, corresponding to an older object from Markov chains, the fundamental matrix [Kemeny and Snell, 1960, Brémaud, 1999, Grinstead and Snell, 1997]. For successor representations in continuous spaces, a finite number of features ϕ are specified first; this can be used for generalization within a family of tasks, e.g., Barreto et al. [2017], Zhang et al. [2017], Grimm et al. [2019], Hansen et al. [2019]. Blier et al. [2021] moves from successor features to successor states by providing pointwise occupancy map estimates even in continuous spaces, without using the sparse reward $\mathbb{1}_{s_t=s'}$. We borrow a successor state learning algorithm from Blier et al. [2021]. Blier et al. [2021] also introduced simpler versions of F and B for a single, fixed policy; Blier et al. [2021] does not consider the every-optimal-policy setting.

There is a long literature on goal-oriented RL. For instance, Schaul et al. [2015] learn goal-dependent value functions. Goal-dependent value functions have been investigated in earlier works such as Foster and Dayan [2002] and Sutton et al. [2011]. Hindsight experience replay (HER) [Andrychowicz et al., 2017] improve the sample efficiency of multiple goal learning with sparse rewards. A family of rewards has to be specified beforehand, such as reaching arbitrary target states. Specifying rewards a posteriori is not possible: for instance, learning to reach target states does not extend to reaching the nearest among several goals, reaching a goal while avoiding forbidden states, or maximizing any dense reward.

Hierarchical methods such as options [Sutton et al., 1999b] can be used for multi-task RL problems. However, RL training and planning on top of the options is still needed after the task is known.

For finite state spaces, Jin et al. [2020a] use reward-free interactions to build a training set that summarizes a finite environment, in the sense that any optimal policies later computed on this training set instead of the true environment are provably ϵ-optimal, for any reward. They prove tight bounds on the necessary set size. Planning still has to be done afterwards for each reward.

7.8 Extended Results: Approximate Solutions and General Goals

This section provides additional theoretical insights and formal statements on approximate FB representations. All the proofs are provided in the appendix E.1. It is organized as follows:

- Section 7.8.1 formalizes the forward-backward representation with a goal or feature space.
- Section 7.8.2 establishes the existence of exact FB representations in finite spaces, and discusses the influence of the dimension d.
- Section 7.8.3 shows how approximate solutions provide approximately optimal policies.
- Section 7.8.4 shows how F and B are successor and predecessor features of each other, and how the policies are optimal for rewards linearly spanned by B.
- Section 7.8.5 explains how to estimate z_R at test time from a state distribution different from the training distribution.
- Section 7.8.6 presents a note of the measure M^π and its density m^π.

Notation. *In general, we denote by M^π the successor measure of policy π as defined in (7.3.1), and by m^π its density, if it exists, with respect to a reference measure ρ. Namely,*

$$M^\pi(s_0, a_0, \mathrm{d}s, \mathrm{d}a) = m^\pi(s_0, a_0, s, a)\rho(\mathrm{d}s, \mathrm{d}a). \tag{7.8.1}$$

Thus, the defining property of forward-backward representations (Definition 4) is $F(s_0, a_0, z)^\top B(s, a) = m^{\pi_z}(s_0, a_0, s, a)$.

We use the same convention for parametric models, with M^π_θ a measure and m^π_θ its density. The reference measure ρ is fixed and may be unknown (typically ρ is the distribution of state-actions in a training set or under an exploration policy).

7.8.1 The Forward-Backward Representation With a Goal or Feature Space

Here we state a generalization of Theorem 1 covering some extensions mentioned in the text.

First, this covers rewards known to only depend on certain features $g = \phi(s, a)$ of the state-action (s, a), where ϕ is a known function with values in some goal state G (for instance, rewards depending only on some components of the state). Then it is enough to compute B as a function of the goal g. Theorem 1 corresponds to $\phi = \mathrm{Id}$. This is useful to introduce prior information when available, resulting in a smaller model (F, B).

This also recovers successor features as in Borsa et al. [2018], defined by user-provided features ϕ. Indeed, fixing B to Id and setting our ϕ to the ϕ of Borsa et al. [2018] (or fixing B to their ϕ and our ϕ to Id) will represent the same set of rewards and policies as in Borsa et al. [2018], namely, optimal policies for rewards linear in ϕ (although with a slightly different learning algorithm and up to a linear change of variables for F and z given by the covariance of ϕ, see Appendix 7.8.4). More generally, keeping the same ϕ but letting B free (with larger d) can provide optimal policies for rewards that are arbitrary functions of ϕ, linear or not.

For this, we extend successor state measures to values in goal spaces, representing the discounted time spent at each goal by the policy. Namely, given a policy π, let M^π be the the successor state measure of π over goals g:

$$M^\pi(s, a, \mathrm{d}g) \triangleq \sum_{t \geq 0} \gamma^t \Pr\left(\phi(s_t, a_t) \in \mathrm{d}g \mid s_0 = s,\, a_0 = a,\, \pi\right) \tag{7.8.2}$$

for each state-action (s, a) and each measurable set $\mathrm{d}g \subset G$. This will be the object approximated by $F(s, a, z)^\top B(g)$.

Second, we use a more general model of successor states: instead of $m \approx F^\top B$ we use $m \approx F^\top B + \bar{m}$ where $\bar{m}$ does not depend on the action, so that the $F^\top B$ part only computes advantages. This lifts the constraint that the model of m has rank at most d, because there is no restriction on the rank on $\bar{m}$: the rank restriction only applies to the advantage function.

Third, we state a form of policy improvement for the FB representation. Namely, the Q-function $F(s, a, z_R)^\top z_R$ for a given reward can be computed as a supremum over all values of z.

For simplicity we state the result with deterministic rewards, but this extends to stochastic rewards, because the expectation z_R will be the same.

Definition 5 (Extended forward-backward representation of an MDP)**.** *Consider an MDP with state space S and action space A. Let $\phi\colon S \times A \to G$ be a function from state-actions to some goal space $G = \mathbb{R}^k$.*

Let $Z = \mathbb{R}^d$ be some representation space. Let

$$F\colon S \times A \times Z \to Z, \qquad B\colon G \to Z, \qquad \bar{m}\colon S \times Z \times G \to \mathbb{R} \tag{7.8.3}$$

be three functions. For each $z \in Z$, define the policy

$$\pi_z(a|s) \triangleq \operatorname*{argmax}_a F(s,a,z)^\top z. \tag{7.8.4}$$

Let ρ be any measure over the goal space G.

We say that F, B, and $\bar{m}$ are an extended forward-backward representation *of the MDP with respect to ρ, if the following holds: for any $z \in Z$, any state-actions (s,a), and any goal $g \in G$, one has*

$$M^{\pi_z}(s,a,\mathrm{d}g) = \Big(F(s,a,z)^\top B(g) + \bar{m}(s,z,g)\Big)\,\rho(\mathrm{d}g) \tag{7.8.5}$$

where M^{π_z} is the successor state measure (7.8.2) of policy π_z.

Theorem 2 (Forward-backward representation of an MDP, with features as goals)**.** *Consider an MDP with state space S and action space A. Let $\phi\colon S \times A \to G$ be a function from state-actions to some goal space $G = \mathbb{R}^k$.*

Let F, B, and $\bar{m}$ be an extended forward-backward representation of the MDP with respect to some measure ρ over G.

Then the following holds. Let $R\colon S \times A \to \mathbb{R}$ be any bounded reward function, and assume that this reward function depends only on $g = \phi(s,a)$, namely, that there exists a function $r\colon G \to \mathbb{R}$ such that $R(s,a) = r(\phi(s,a))$. Set

$$z_R \triangleq \int_{g \in G} r(g) B(g)\,\rho(\mathrm{d}g) \tag{7.8.6}$$

assuming the integral exists.

Then:

1. *π_{z_R} is an optimal policy for reward R in the MDP.*
2. *For any $z \in Z$, the Q-function of policy π_z for the reward R is equal to*

$$Q^{\pi_z}(s,a) = F(s,a,z)^\top z_R + \bar{V}^z(s) \tag{7.8.7}$$

and the optimal Q-function $Q^\star_R$ is obtained when $z = z_R$:

$$Q^\star_R(s,a) = F(s,a,z_R)^\top z_R + \bar{V}^{z_R}(s). \tag{7.8.8}$$

Here

$$\bar{V}^z(s) \triangleq \int_{g \in G} \bar{m}(s,z,g) r(g)\,\rho(\mathrm{d}g) \tag{7.8.9}$$

and in particular $\bar{V} = 0$ if $\bar{m} = 0$.

The advantages $Q^{\pi_z}(s,a) - Q^{\pi_z}(s,a')$ do not depend on $\bar{V}$, so computing $\bar{V}$ is not necessary to obtain the policies.

3. *If $\bar{m} = 0$, then for any state-action (s, a) one has*

$$Q_R^\star(s,a) = F(s,a,z_R)^\top z_R = \sup_{z \in Z} F(s,a,z)^\top z_R. \tag{7.8.10}$$

(We do not claim that $\bar{V}$ is the value function and $F^\top z_R$ the advantage function, only that the sum is the Q-function. When $\bar{m} = 0$, the term $F^\top z_R$ is the whole Q-function.)

The last point of the theorem is a form of policy improvement. Indeed, by the second point, $F(s,a,z)^\top z_R$ is the estimated Q-function of policy π_z for rewards r. This may be useful if z_R falls outside of the training distribution for F: then the values of $F(s,a,z_R)$ may not be safe to use. In that case, it may be useful to use a finite set $Z' \subset Z$ of values of z closer to the training distribution, and use the estimate $\sup_{z \in Z'} F(s,a,z)^\top z_R$ instead of $F(s,a,z_R)^\top z_R$ for the optimal Q-function. A similar option has been used, e.g., in Borsa et al. [2018], but in the end it was not necessary in our experiments.

Remark 3. *Formally, the statement holds for arbitrary ρ, but it only makes sense if ρ has full support (or at least covers all reachable parts of the state space): (7.8.5) requires the support of M^{π_z} to be included in that of ρ. Otherwise, FB representations may not exist and the statement is empty.*

7.8.2 Existence of Exact FB Solutions, Influence of Dimension d, Uniqueness

Existence of exact FB representations in finite spaces. We now prove existence of an exact solution for finite spaces if the representation dimension d is at least $\#S \times \#A$. Solutions are never unique: one may always multiply F by an invertible matrix C and multiply B by $(C^\top)^{-1}$, see Remark 4 below (this allows us to impose orthonormality of B in the experiments).

The constraint $d \geq \#S \times \#A$ can be largely overestimated depending on the tasks of interest, though. For instance, we prove below that in an n-dimensional toric grid $S = \{1, \dots, k\}^n$, $d = 2n$ is enough to obtain optimal policies for reaching every target state (a set of tasks smaller than optimizing all possible rewards).

Proposition 6 (Existence of an exact FB representation for finite state spaces)**.** *Assume that the state and action spaces S and A of an MDP are finite. Let $Z = \mathbb{R}^d$ with $d \geq \#S \times \#A$. Let ρ be any probability distribution on $S \times A$, with $\rho(s,a) > 0$ for any (s,a).*

Then there exists $F \colon S \times A \times Z \to Z$ and $B \colon S \times A \to Z$, such that $F^\top B$ is equal to the successor state density of π_z with respect to ρ:

$$F^\top(s,a,z)B(s',a') = \sum_{t \geq 0} \gamma^t \frac{\Pr((s_t,a_t) = (s',a') \mid s_0 = s,\, a_0 = a,\, \pi_z)}{\rho(s',a')} \tag{7.8.11}$$

for any $z \in Z$ and any state-actions (s,a) and (s',a'), where π_z is defined as in Theorem 1 by $\pi_z(s) = \operatorname{argmax}_a F(s,a,z)^\top z$.

A small dimension d can be enough for navigation: examples. In practice, even a small d can be enough to get optimal policies for reaching arbitrary many states (as opposed to optimizing all possible rewards). Let us give an example with S a toric n-dimensional grid of size k.

Let us start with $n = 1$. Take $S = \{0, \dots, k-1\}$ to be a length-k cycle with three actions $a \in \{-1, 0, 1\}$ (go left, stay in place, go right). Take $d = 2$, so that $Z = \mathbb{R}^2 \simeq \mathbb{C}$.

We consider the tasks of reaching an arbitrary target state s', for every $s' \in S$. Thus the goal state is $G = S$ in the notation of Theorem 2, and B only depends on s'. The policy for such a reward is $\pi_{z_R} = \pi_{B(s')}$.

For a state $s \in \{0, \dots, k-1\}$ and action $a \in \{-1, 0, 1\}$, define

$$F(s,a,z) \triangleq e^{2i\pi(s+a)/k}, \qquad B(s) \triangleq e^{2i\pi s/k}. \tag{7.8.12}$$

Then one checks that $\pi_{B(s')}$ is the optimal policy for reaching s', for every $s' \in S$. Indeed, $F(s,a,z_R)^\top z_R = \cos(2\pi(s+a-s')/k)$. This is maximized for the action a that brings s closer to s'.

So the policies will be optimal for reaching every target $s' \in S$, despite the dimension being only 2.

By taking the product of n copies of this example, this also works on the n-dimensional toric grid $S = \{0, \dots, k-1\}^n$ with $2n+1$ actions (add ± 1 in each direction or stay in place), with a representation of dimension $d = 2n$ in $\mathbb{C}^n$, namely, by taking $B(s)_j \triangleq e^{2i\pi s_j/k}$ for ecah direction j and likewise for F. Then $\pi_{B(s')}$ is the optimal policy for reaching s' for every $s' \in S$.

More generally, if one is only interested in the optimal policies for reaching states, then it is easy to show that there exist functions $F\colon S \times A \to Z$ and $B\colon S \to Z$ such that the policies π_z describe the optimal policies to reach each state: it is enough that B be injective (typically requiring $d = \dim(S)$). Indeed, for any state $s \in S$, let $\pi^\star_s$ be the optimal policy to reach s. We want π_z to be equal to $\pi^\star_s$ for $z = B(s)$ (the value of z_R for a reward located at s). This translates as $\operatorname{argmax}_a F(s',a,B(s))^\top B(s) = \pi^\star_s(s')$ for every other state s'. This is realized just by letting F be any function such that $F(s', \pi^\star_s(s'), B(s)) \triangleq B(s)$ and $F(s',a,B(s)) \triangleq -B(s)$ for every other action a. As soon as B is injective, there exists such a function F.

Let us turn to uniqueness of F and B.

Remark 4. *Let C be an invertible $d \times d$ matrix. Given F and B as in Theorem 1, define*

$$B'(s,a) \triangleq CB(s,a), \qquad F'(s,a,z) \triangleq (C^\top)^{-1} F(s,a,C^{-1}z) \tag{7.8.13}$$

together with the policies $\pi'_z(s) \triangleq \operatorname{argmax}_a F'(s,a,z)^\top z$. *For each reward* r, *define* $z'_R \triangleq \mathbb{E}_{(s,a)\sim\rho}[r(s,a)B'(s,a)]$.

Then this operation does not change the policies or estimated Q-values: for any reward, we have $\pi'_{z'_R} = \pi_{z_R}$, *and* $F'(s,a,z'_R)^\top z'_R = F(s,a,z_R)^\top z_R$.

In particular, assume that the components of B *are linearly independent. Then, taking* $C = \left(\mathbb{E}_{(s,a)\sim\rho} B(s,a)B(s,a)^\top\right)^{-1/2}$, B' *is* $L^2(\rho)$*-orthonormal. So up to reducing the dimension* d *to* $\operatorname{rank}(B)$, *we can always assume that* B *is* $L^2(\rho)$*-orthonormal.*

Reduction to orthonormal B will be useful in some proofs below. Even after imposing that B be orthonormal, solutions are not unique, as one can still apply a rotation matrix on the variable z.

For a single policy π_z, the $F^\top B$ decomposition may be further standardized in several ways: after a linear change of variables in $\mathbb{R}^d$, and up to decreasing d by removing unused directions in $\mathbb{R}^d$, one may assume either that $\operatorname{Cov}_\rho B = \mathrm{Id}$ and $\operatorname{Cov}_\rho F$ is diagonal, or that $\operatorname{Cov}_\rho F = \operatorname{Cov}_\rho B$ is diagonal, or write the decomposition as $\tilde{F}^\top D\tilde{B}$ with D diagonal and $\operatorname{Cov}_\rho \tilde{F} = \operatorname{Cov}_\rho \tilde{B} = \mathrm{Id}$, thus corresponding to an approximation of the singular value decomposition of the successor measure M^{π_z} in $L^2(\rho)$. However, since B is shared between all values of z and all policies π_z, it is a priori not possible to realize this for all z simultaneously.

7.8.3 Approximate Solutions Provide Approximately Optimal Policies

Here we prove that the optimality in Theorems 1 and 2 is robust to approximation errors during training: approximate solutions still provide approximately optimal policies. We deal first with the approximation errors on (F,B) during unsupervised training, then on z_R during the reward estimation phase (in case the reward is not known explicity).

Influence of Approximate F and B: Optimality Gaps are Proportional to $m^\pi - F^\top B$

In continuous spaces, Theorems 1 and 2 are somewhat spurious: the equality $F^\top B = m$ will never hold exactly with finite representation dimension d. Instead, $F^\top B$ will only be a rank-d approximation of m. Even in finite spaces, since F and B are learned by a neural network, we can only expect that $F^\top B \approx m$ in general. Therefore, we provide results that extend Theorems 1 and 2 to approximate training.

Optimality gaps are directly controlled by the error $m^\pi - F^\top B$ on the solution $F^\top B$. We provide this result for different notions of approximations between m^π and $F^\top B$. First, in sup norm over (s,a) but in expectation over (s',a') (so that a

perfect model of the successor states (s', a') of each (s, a) is not necessary, only an average model). Second, for the weak topology on measures (this is the most relevant in continuous spaces: for instance, a Dirac measure can be approached by a continuous model in the weak topology). Finally, we provide pointwise estimates instead of only norms: for any reward, we show that the optimality gaps at each state can be bounded by an explicit matrix product directly involving the FB error matrix $m^\pi - F^\top B$ (Theorem 6).

F and B are trained such that $F(s, a, z)^\top B(s', a')$ approximates the successor state density $m^{\pi_z}(s, a, s', a')$. In the simplest case, we prove that if for some reward R,

$$\mathbb{E}_{(s',a')\sim\rho} \left| F(s, a, z_R)^\top B(s', a') - m^{\pi_{z_R}}(s, a, s', a') \right| \leq \epsilon \tag{7.8.14}$$

for every (s, a), then the optimality gap of policy π_{z_R} is at most $(3\epsilon/(1-\gamma))\sup|R|$ for that reward (Theorem 5, first case).

In continuous spaces, m^π is usually a distribution (Appendix 7.8.6), so such an approximation will not hold, and it is better to work on the measures themselves rather than their densities, namely, to compare $F^\top B\rho$ to M^π instead of $F^\top B$ to m^π. We prove that if $F^\top B\rho$ is close to M^π in the weak topology, then the resulting policies are optimal for any Lipschitz reward.[2]

Remember that a sequence of nonnegative measures μ_n converges weakly to μ if for any bounded, continuous function f, $\int f(x)\mu_n(dx)$ converges to $\int f(x)\mu(\mathrm{d}x)$ (Bogachev [2007], §8.1). The associated topology can be defined via the following *Kantorovich–Rubinstein* norm on nonnegative measures (Bogachev [2007], §8.3)

$$\|\mu - \mu'\|_{\mathrm{KR}} \triangleq \sup\left\{ \left| \int f(x)\mu(\mathrm{d}x) - \int f(x)\mu'(\mathrm{d}x) \right| : f \text{ 1-Lipschitz function with } \sup|f| \leq 1 \right\} \tag{7.8.15}$$

where we have equipped the state-action space with any metric compatible with its topology.[3]

The following theorem states that if $F^\top B$ approximates the successor state density of the policy π_z (for various sorts of approximations), then π_z is approximately optimal. Given a reward function r on state-actions, we denote

$$\|r\|_\infty \triangleq \sup_{(s,a)\in S\times A} |r(s, a)| \tag{7.8.16}$$

[2]This also holds for continuous rewards, but the Lipschitz assumption yields an explicit bound in Theorem 5.

[3]The Kantorovich–Rubinstein norm is closely related to the L^1 Wasserstein distance on probability distributions, but slightly more general as it does not require the distance functions to be integrable: the Wasserstein distance metrizes weak convergence among those probability measures such that $\mathbb{E}[d(x, x_0)] < \infty$.

and

$$\|r\|_{\text{Lip}} \triangleq \sup_{(s,a)\neq(s',a')\in S\times A} \frac{r(s,a)-r(s',a')}{d((s,a),(s',a'))} \tag{7.8.17}$$

where we have chosen any metric on state-actions.

The first statement is for any bounded reward. The second statement only assumes an $F^\top B$ approximation in the weak topology but only applies to Lipschitz rewards. The third statement is more general and is how we prove the first two: weaker assumptions on $F^\top B$ work on a stricter class of rewards.

Theorem 5 (If F and B approximate successor states, then the policies π_z yield approximately optimal returns)**.** *Let $F\colon S\times A\times Z\to Z$ and $B\colon S\times A\to Z$ be any functions, and define the policy $\pi_z(s)=\operatorname{argmax}_a F(s,a,z)^\top z$ for each $z\in Z$.*

Let ρ be any positive probability distribution on $S\times A$, and for each policy π, let m^π be the density of the successor state measure M^π of π with respect to ρ. Let

$$\hat{m}^z(s,a,s',a') \triangleq F(s,a,z)^\top B(s',a'), \qquad \hat{M}^z(s,a,\mathrm{d}s',\mathrm{d}a') \triangleq \hat{m}^z(s,a,s',a')\rho(\mathrm{d}s',\mathrm{d}a') \tag{7.8.18}$$

be the estimates of m and M obtained via the model F and B.

Let $r\colon S\times A\to\mathbb{R}$ be any bounded reward function. Let $V^\star$ be the optimal value function for this reward r. Let $\hat{V}^{\pi_z}$ be the value function of policy π_z for this reward. Let $z_R=\mathbb{E}_{(s,a)\sim\rho}[r(s,a)B(s,a)]$.

Then:

1. *If $\mathbb{E}_{(s',a')\sim\rho}\left|\hat{m}^{z_R}(s,a,s',a')-m^{\pi_{z_R}}(s,a,s',a')\right|\le\epsilon$ for any (s,a) in $S\times A$, then $\left\|\hat{V}^{\pi_{z_R}}-V^\star\right\|_\infty\le 3\epsilon\|r\|_\infty/(1-\gamma)$.*

2. *If r is Lipschitz and $\left\|\hat{M}^{z_R}(s,a,\cdot)-M^{\pi_{z_R}}(s,a,\cdot)\right\|_{\text{KR}}\le\epsilon$ for any $(s,a)\in S\times A$, then $\left\|\hat{V}^{\pi_{z_R}}-V^\star\right\|_\infty\le 3\epsilon\max(\|r\|_\infty,\|r\|_{\text{Lip}})/(1-\gamma)$.*

3. *More generally, let $\|\cdot\|_A$ be a norm on functions and $\|\cdot\|_B$ a norm on measures, such that $\int f\,\mathrm{d}\mu\le\|f\|_A\|\mu\|_B$ for any function f and measure μ. Then for any reward function r such that $\|r\|_A<\infty$,*

$$\left\|\hat{V}^{\pi_{z_R}}-V^\star\right\|_\infty\le\frac{3\|r\|_A}{1-\gamma}\sup_{s,a}\left\|\hat{M}^{z_R}(s,a,\cdot)-M^{\pi_{z_R}}(s,a,\cdot)\right\|_B. \tag{7.8.19}$$

Moreover, the optimal Q-function is close to $F(s,a,z_R)^\top z_R$:

$$\sup_{s,a}\left|F(s,a,z_R)^\top z_R-Q^\star(s,a)\right|\le\frac{2\|r\|_A}{1-\gamma}\sup_{s,a}\left\|\hat{M}^{z_R}(s,a,\cdot)-M^{\pi_{z_R}}(s,a,\cdot)\right\|_B. \tag{7.8.20}$$

Pointwise optimality gaps. We now turn to a more precise estimation of the optimality gap at each state, expressed directly as a matrix product involving the FB error $m^\pi - F^\top B$.

Here we assume that the state space is finite: this is not essential but simplifies notation since everything can be represented as matrices and vectors. Rcall that for each stochastic policy π, P^π denotes the associated stochastic matrix on state-actions: $(P_\pi)_{(sa)(s'a')} = P(s'|s,a)\pi(s')[a']$. We also view rewards and Q-functions as vectors indexed by state-actions.

Theorem 6. *Assume that the state space is finite. Let $F\colon S \times A \times \mathbb{R}^d \to \mathbb{R}^d$ and $B\colon S \times A \to \mathbb{R}^d$ be any functions. Define π_z as in Theorem 1. Let $\rho > 0$ be any distribution over state-actions, and let m^π be the successor density (7.8.1) of policy π with respect to ρ.*

For each $z \in \mathbb{R}^d$, define the FB error $E(z)$ as a matrix over state-actions:

$$E(z)_{(sa)(s'a')} \triangleq m^{\pi_z}(s,a,s',a') - F(s,a,z)^\top B(s',a'). \tag{7.8.21}$$

Let r be any reward function and let $Q^\star$ and $\pi^\star$ be its optimal Q-function and policy. Let $z_R = \mathbb{E}_\rho[r.B]$ as in Theorem 1, and let $Q^{\pi_{z_R}}$ be the Q-function of policy π_{z_R}.

Then we have the componentwise inequality between state-action vectors

$$0 \le Q^\star - Q^{\pi_{z_R}} \le \left(\sum_{t\ge 0} \gamma^{t+1}(P^{\pi^\star})^t\right)\left(P^{\pi^\star} - P^{\pi_{z_R}}\right) E(z_R)\,\mathrm{diag}(\rho)\, r. \tag{7.8.22}$$

In particular, if $E(z_R) = 0$ then π_{z_R} is optimal for reward r.

The matrix $\sum_{t\ge 0} \gamma^{t+1}(P^{\pi^\star})^t$ represents states visited along the optimal trajectory starting at the initial state. Multiplying by $(P^{\pi^\star} - P^{\pi_{z_R}})$ visits states one step away from this trajectory. Therefore, what matters are the values of the FB error matrix $E(z_R)_{(sa)(s'a')}$ at state-actions (s,a) one step away from the optimal trajectories.

In unsupervised RL, we have no control over the first part $\sum_{t\ge 0} \gamma^{t+1}(P^{\pi^\star})^t$, which depends only on the optimal trajectories of the unknown future tasks r. Therefore, in general it makes sense to just minimize $E(z)$ in some matrix norm, for as many values of z as possible. (The choice of matrix norm influences which rewards will be best optimized, as illustrated by Theorem 5.)

An approximate z_R yields an approximately optimal policy

We now turn to the second source of approximation: computing z_R in the reward estimation phase. This is a problem only if the reward is not specified explicitly.

We deal in turn with the effect of using a model of the reward function, and the effect of estimating $z_R = \mathbb{E}_{(s,a)\sim\rho} r(s,a)B(s,a)$ via sampling.

Which of these options is better depends a lot on the situation. If training of F and B is perfect, by construction the policies are optimal for each z_R: thus, estimating z_R from rewards sampled at N states $(s_i, a_i) \sim \rho$ will produce the optimal policy for exactly that empirical reward, namely, a nonzero reward at each (s_i, a_i) but zero everywhere else, thus overfitting the reward function. Reducing the dimension d reduces this effect, since rewards are projected on the span of the features in B: B plays both the roles of a transition model and a reward regularizer. This appears as a $\sqrt{d/N}$ factor in Theorem 7 below.

Thus, if both the number of samples to train F and B and the number of reward samples are small, using a smaller d will regularize both the model of the environment and the model of the reward. However, if the number of samples to train F and B is large, yielding an excellent model of the environment, but the number of reward samples is small, then learning a model of the reward function will be a better option than direct empirical estimation of z_R.

The first result below states that reward misidentification comes on top of the approximation error of the $F^\top B$ model. This is relevant, for instance, if a reward model $\hat{r}$ is estimated by an external model using some reward values.

Proposition 7 (Influence of estimating z_R by an approximate reward)**.** *Let $\hat{r}\colon S \times A \to \mathbb{R}$ be any reward function. Let $\hat{z}_R = \mathbb{E}_{(s,a)\sim\rho}[\hat{r}(s,a)B(s,a)]$.*

Let ϵ_{FB} be the error attained by the $F^\top B$ model in Theorem 5 for reward $\hat{r}$; namely, assume that $\left\|\hat{V}^{\pi_{\hat{z}_R}} - \hat{V}^\star\right\|_\infty \leq \epsilon_{FB}$ with $\hat{V}^\star$ the optimal value function for $\hat{r}$.

Then the policy $\pi_{\hat{z}_R}(s) = \operatorname{argmax}_a F(s,a,\hat{z}_R)^\top \hat{z}_R$ defined by the model $\hat{r}$ is $\left(\dfrac{2\left\|r-\hat{r}\right\|_\infty}{(1-\gamma)} + \epsilon_{FB}\right)$-optimal for reward r.

This still assumes that the expectation $\hat{z}_R = \mathbb{E}_{(s,a)\sim\rho}[\hat{r}(s,a)B(s,a)]$ is computed exactly for the model $\hat{r}$. If $\hat{r}$ is given by an explicit model, this expectation can in principle be computed on the whole replay buffer used to train F and B, so variance would be low. Nevertheless, we provide an additional statement which covers the influence of the variance of the estimator of z_R, whether this estimator uses an external model $\hat{r}$ or a direct empirical average reward observations $r(s,a)B(s,a)$.

Definition 6. *The* skewness $\zeta(B)$ *of B is defined as follows. Assume B is bounded. Let $B_1, \ldots, B_d\colon S \times A \to \mathbb{R}$ be the functions of (s,a) defined by each component of B. Let $\langle B \rangle$ be the linear span of the $(B_i)_{1\leq i\leq d}$ as functions on $S \times A$. Set*

$$\zeta(B) \triangleq \sup_{f\in\langle B\rangle, f\neq 0} \frac{\|f\|_\infty}{\|f\|_{L^2(\rho)}}. \tag{7.8.23}$$

Theorem 7 (Influence of estimating z_R by empirical averages)**.** *Assume that* $z_R = \mathbb{E}_{(s,a)\sim\rho}[r(s,a)B(s,a)]$ *is estimated via*

$$\hat{z}_R \triangleq \frac{1}{N}\sum_{i=1}^{N} \hat{r}_i B(s_i, a_i) \tag{7.8.24}$$

using N *independent samples* $(s_i, a_i) \sim \rho$*, where the* r_i *are random variables such that* $\mathbb{E}[\hat{r}_i|s_i, a_i] = r(s_i, a_i)$*,* $\mathrm{Var}[\hat{r}_i|s_i, a_i] \leq v$ *for some* $v \in \mathbb{R}$*, and the* $\hat{r}_i$ *are mutually independent given* $(s_i, a_i)_{i=1,\dots,N}$*.*

Let $V^\star$ *be the optimal value function for reward* r*, and let* $\hat{V}$ *be the value function of the estimated policy* $\pi_{\hat{z}_R}$ *for reward* r*.*

Then, for any $\delta > 0$*, with probability at least* $1 - \delta$*,*

$$\left\|\hat{V} - V^\star\right\|_\infty \leq \epsilon_{FB} + \frac{2}{1-\gamma}\sqrt{\frac{\zeta(B)\, d}{N\delta}\left(v + \left\|r(s,a) - \mathbb{E}_\rho r\right\|^2_{L^2(\rho)}\right)} \tag{7.8.25}$$

which is therefore the bound on the optimality gap of $\pi_{\hat{z}_R}$ *for* r*. Here* ϵ_{FB} *is the error due to the* $F^\top B$ *model approximation, defined as in Proposition 7.*

The proofs are not direct, because F is not continuous with respect to z. Contrary to Q-values, successor states are not continuous in the reward: if an action has reward 1 and the reward for another action changes from $1-\epsilon$ to $1+\epsilon$, the return values change by at most 2ϵ, but the actions and states visited by the optimal policy change a lot. So it is not possible to reason by continuity on each of the terms involved.

7.8.4 F and B as Successor and Predecessor Features of Each Other, Optimality for Rewards in the Span of B

We now give two statements. The first encodes the idea that F encodes the future of a state while B encodes the past of a state.

The second proves that the FB policies are optimal for any reward that lies in the linear span of the features learned in B; for rewards out of this span, it is best if the features in B are spatially smooth.

The intuition that F and B encode the future and past of states is formalized as follows: if F and B minimize their unsupervised loss, then F is equal to the successor features from the dual features of B, and B is equal to the *predecessor* features from the dual features of F (Theorem 8).

This statement holds for a fixed z and the corresponding policy π_z. So, for the rest of this section, z is fixed.

By "dual" features we mean the following. Define the $d \times d$ covariance matrices

$$\mathrm{Cov}\, F \triangleq \mathbb{E}_{(s,a)\sim\rho}[F(s,a,z)F(s,a,z)^\top], \qquad \mathrm{Cov}\, B \triangleq \mathbb{E}_{(s,a)\sim\rho}[B(s,a)B(s,a)^\top]. \tag{7.8.26}$$

Then $(\mathrm{Cov}\, F)^{-1/2} F(s,a,z)$ is $L^2(\rho)$-orthonormal and likewise for B. The "dual" features are $(\mathrm{Cov}\, F)^{-1} F(s,a,z)$ and $(\mathrm{Cov}\, B)^{-1} B(s,a)$, without the square root: these are the least square solvers for F and B respectively, and these are the ones that appear below.

The unsupervised FB loss for a fixed z is

$$\ell(F,B) \triangleq \int \left| F(s,a,z)^\top B(s',a') - \sum_{t\geq 0} \gamma^t \frac{P_t(\mathrm{d}s', \mathrm{d}a' | s, a, \pi_z)}{\rho(\mathrm{d}s', \mathrm{d}a')} \right|^2 \rho(\mathrm{d}s, \mathrm{d}a) \rho(\mathrm{d}s', \mathrm{d}a') \tag{7.8.27}$$

$$= \left\| F(\cdot, z)^\top B(\cdot) - m^{\pi_z}(\cdot, \cdot) \right\|^2_{L^2(\rho)\otimes L^2(\rho)} . \tag{7.8.28}$$

Thus, minimizers in dimension d correspond to an SVD of the successor state density in $L^2(\rho)$, truncated to the largest d singular values.

Theorem 8. *Consider a smooth parametric model for F and B, and assume this model is overparameterized.* [4] *Also assume that the data distribution ρ has positive density everywhere.*

Let $z \in Z$. Assume that for this z, F and B lie in $L^2(\rho)$ and achieve a local extremum of $\ell(F,B)$ within this parametric model. Namely, the derivative $\partial_\theta \ell(F,B)$ of the loss with respect to the parameters θ of F is 0, and likewise for B.

Then F is equal to $(\mathrm{Cov}\, B)^{-1}$ times the successor features of B: for any $(s,a) \in S \times A$,

$$(\mathrm{Cov}\, B) F(s,a,z) = \sum_{t\geq 0} \gamma^t \int_{(s',a')} P_t(\mathrm{d}s', \mathrm{d}a' | s, a, \pi_z)\, B(s',a') \tag{7.8.29}$$

and B is equal to $(\mathrm{Cov}\, F)^{-1}$ times the predecessor features of F:

$$(\mathrm{Cov}\, F) B(s',a') = \sum_{t\geq 0} \gamma^t \int_{(s,a)} \frac{P_t(\mathrm{d}s', \mathrm{d}a' | s, a, \pi_z)}{\rho(\mathrm{d}s', \mathrm{d}a')}\, F(s,a,z)\, \rho(\mathrm{d}s, \mathrm{d}a) \tag{7.8.30}$$

ρ-almost everywhere. Here the covariances have been defined in (7.8.26), and $P_t(\cdot|s,a,\pi)$ denotes the law of (s_t, a_t) under trajectories starting at (s,a) and following policy π.

The same result holds when working with features $\phi(s',a')$, just by applying it to $B \circ \phi$.

[4] Intuitively, a parametric function f is overparameterized if every possible small change of f can be realized by a small change of the parameter. Formally, we say that a parametric family of functions $\theta \in \Theta \mapsto f_\theta \in L^2(X, \mathbb{R}^d)$ smoothly parameterized by θ, on some space X, is *over-parameterized* if, for any θ, the differential $\partial_\theta f_\theta$ is surjective from Θ to $L^2(X, \mathbb{R}^d)$. For finite X, this implies that the dimension of θ is larger than $\#X$. For infinite X, this implies that $\dim(\theta)$ is infinite, such as parameterizing functions on $[0;1]$ by their Fourier expansion.

Note that in the FB framework, we may normalize either F or B (Remark 4), but not both.

As a consequence of Theorem 8, we characterize below which kind of rewards we can capture if we fix B and train only F to minimize the unsupervised loss given B. Namely, we show that for any reward r, the resulting policy is optimal for the $L^2(\rho)$-orthogonal projection of r onto the span of B.

Theorem 9 (Optimizing F for a given B; influence of the span of B)**.** *Let B a fixed function in $L^2(\rho, \mathbb{R}^d)$. Define the* span *of B as the set of functions $w^\top B \in L^2(\rho, \mathbb{R})$ when w ranges in $\mathbb{R}^d$.*

Consider a smooth parametric model for F, and assume this model is overparameterized. Assume that F lies in $L^2(\rho)$ and achieves a local extremum of $\ell(F, B)$ within this parametric model for each $z \in Z$.

Assume that the data distribution ρ has positive density everywhere.

Then for any bounded reward function $r\colon S \times A \to \mathbb{R}$ that lies in the span of B, π_{z_R} is an optimal policy for the reward r.

More generally, for any bounded reward function $r\colon S \times A \to \mathbb{R}$, the policy π_{z_R} is an optimal policy for the reward r_B defined as the $L^2(\rho)$-orthogonal projection of r onto the span of B.

Moreover,

$$\|V^{\pi_{z_R}} - V^\star\|_\infty \leq \frac{2}{1-\gamma} \|r - r_B\|_\infty \tag{7.8.31}$$

with $V^\star$ the optimal value function for r. More precisely,

$$\|V^{\pi_{z_R}} - V^\star\|_\infty \leq \left\|(\mathrm{Id} - \gamma P_{\pi^\star})^{-1}(r - r_B)\right\|_\infty + \left\|(\mathrm{Id} - \gamma P_{\pi_{z_R}})^{-1}(r - r_B)\right\|_\infty \tag{7.8.32}$$

with $\pi^\star$ the optimal policy for reward r, and notation P_π as in Theorem 6.

The last bounds implies the previous one, as $(\mathrm{Id} - \gamma P_\pi)^{-1}$ is bounded by $\frac{1}{1-\gamma}$ in L^∞ norm for any policy π.

Discussion: optimal B and priors on rewards. This bound is interesting if $r - r_B$ is small, namely, if B captures most of the components of the reward functions we are interested in. But even for rewards not spanned by B, the bound is smaller if $r - r_B$ avoids the largest eigendirections of $(\mathrm{Id} - P_\pi)^{-1}$ for various policies π, namely, if B captures these largest eigendirections. These eigendirections are those of P_π: so the bound will be small if B contains the largest eigendirections of P_π for various policies π, corresponding to spatially continuity of functions under transitions in the environment ($P_\pi f$ close to f).

If we are interested in spatially smooth rewards r, then $r - r_B$ is small if B if captures smooth functions first. But even for rewards not spanned by B, and for non-spatially smooth rewards (e.g., goal-oriented problems with reward

$\mathbb{1}_{s=\text{goal}}$), the bound (7.8.32) shows that B should first capture spatially smooth eigenvectors of many policies π.

Is this a natural consequence of FB training? Up to some approximations, yes. For a single z and policy π_z, the loss (7.8.27) used to train B is optimal when B captures the largest *singular* directions of $(\text{Id} - P_{\pi_z})^{-1}$, which is slightly different. The optimal policy $(\text{Id} - P_{\pi^\star})^{-1}$ is not represented in the criterion, and it is not clear to what extent spatial continuity with respect to P_{π_z} or $P_{\pi^\star}$ differ. Moreover, in the full algorithm, B is shared between several z and several policies. So we have no rigorous result here. Still, the intuition shows that FB training goes in the correct direction.

7.8.5 Estimating z_R from a Different State Distribution at Test Time

Theorem 1 requires $z_R = \mathbb{E}_{(s,a)\sim\rho}[r(s,a)B(s,a)]$ to be estimated using rewards observed from the same state-action distribution ρ as the one used to define m^π and train F and B. The algorithm of Section 7.5 computes m^π with respect to the distribution ρ of the training set. So in general, estimating z_R requires either being able to run the exploration policy again once reward samples are available, or being able to explicitly estimate $r(s,a)$ on states stored in the training set.

However, at test time, we would generally like to use the policy learned, rather than the exploration policy. This will result in a distribution of states-actions $\rho_{\text{test}}(s,a)$ different from the training distribution $\rho(s,a)$.

If the training set remains accessible, a generic solution is to train a model $\hat{r}(s,a)$ of the reward function, from rewards $r(s,a)$ observed at test time under any distribution $(s,a) \sim \rho_{\text{test}}$. Then one can estimate z_R by averaging this model $\hat{r}$ over state-actions sampled from the training set:

$$\hat{z}_R \triangleq \mathbb{E}_{(s,a)\sim\rho}[\hat{r}(s,a)B(s,a)]. \tag{7.8.33}$$

However, the training set may not be available anymore at test time. But as it turns out, if we use for $\hat{r}$ a *linear* model based on the features learned in B, then we do not need to store the training set: it is enough to estimate the matrix $\text{Cov}_\rho(B)$, which can be pre-computed during training. This is summarized in the following.

Proposition 8. *Let $\hat{r}$ be the linear model of rewards computed at test time by linear regression of the reward r over the components $B_1, \dots, B_d$ of B, with state-actions taken from a test distribution ρ_{test}. Explicitly,*

$$\hat{r}(s,a) \triangleq B(s,a)^\top w, \qquad w \triangleq \left(\text{Cov}_{\rho_{\text{test}}} B\right)^{-1} \mathbb{E}_{(s,a)\sim\rho_{\text{test}}}[r(s,a)B(s,a)]. \tag{7.8.34}$$

Here we assume that $\mathrm{Cov}_{\rho_{\text{test}}} B = \mathbb{E}_{(s,a)\sim\rho_{\text{test}}}[B(s,a)B(s,a)^\top]$ *is invertible.*

Then the estimate $\hat{z}_R$ *computed by using this model* $\hat{r}$ *in* (7.8.33) *is*

$$\hat{z}_R = (\mathrm{Cov}_\rho B)\left(\mathrm{Cov}_{\rho_{\text{test}}} B\right)^{-1} \mathbb{E}_{(s,a)\sim\rho_{\text{test}}}[r(s,a)B(s,a)]. \tag{7.8.35}$$

Moreover, if $\rho = \rho_{\text{test}}$*, or if* r *is linear over* B*, then* $\hat{z}_R = z_R$ *.*

For this estimate, $\mathrm{Cov}_{\rho_{\text{test}}} B$ and $\mathbb{E}_{(s,a)\sim\rho_{\text{test}}}[r(s,a)B(s,a)]$ can be computed at test time, while the matrix $\mathrm{Cov}_\rho B$ must be computed at training time. This way, the training set can be discarded.

If the estimate (7.8.34) is used, the learned policies correspond to using universal successor features approximators Borsa et al. [2018] on top of the features learned by B. Indeed, universal successor features with features ϕ use the policies $\mathrm{argmax}_a \psi(s,a,w)^\top w$ with $w = \left(\mathrm{Cov}_{\rho_{\text{test}}} \phi\right)^{-1} \mathbb{E}_{(s,a)\sim\rho_{\text{test}}}[r(s,a)\phi(s,a)]$ the regression vector of r over the features ϕ, and with $\psi = \mathrm{Succ}(\phi)$ the successor features of ϕ. Here we use the policies $\pi_{\hat{z}_R} = \mathrm{argmax}_a F(s,a,\hat{z}_R)^\top \hat{z}_R$. Let us set $\phi \triangleq B$ as the base features in successor features. Then the above shows that $\hat{z}_R = (\mathrm{Cov}_\rho B)w$ when using the linear model of rewards. Moreover, we proved in Theorem 8 that the optimum for F is $F = (\mathrm{Cov}_\rho B)^{-1}\mathrm{Succ}(B)$. Therefore, the policies coincide in this situation.

Thus, universal successor features based on B appear as a particular case if a linear model of rewards is used at test time, although in general any reward model may be used in (7.8.33).

7.8.6 A Note on the Measure M^π and its Density m^π

In finite spaces, the definition of the successor state density m^π via

$$M^\pi(s,a,\mathrm{d}s',\mathrm{d}a') = m^\pi(s,a,s',a')\rho(\mathrm{d}s',\mathrm{d}a') \tag{7.8.36}$$

with respect to the data distribution ρ poses no problem, as long as the data distribution is positive everywhere.

In continuous spaces, this can be understood as the (Radon–Nikodym) density of M^π with respect to ρ, assuming M^π has no singular part with respect to ρ. However, this is *never* the case: in the definition (7.3.1) of the successor state measure M^π, the term $t = 0$ produces a Dirac measure $\delta_{s,a}$. So M^π has a singular component due to $t = 0$, and m^π is better thought of as a distribution.

When m^π is a distribution, a continuous parametric model m^π_θ learned by (7.5.2) can approximate m^π in the weak topology only: $m^\pi_\theta \rho$ approximates M^π for the weak convergence of measures. Thus, for the forward-backward representation, $F(s,a,z)^\top B(s',a')\rho(\mathrm{d}s',\mathrm{d}a')$ weakly approximates $M^{\pi_z}(s,a,\mathrm{d}s',\mathrm{d}a')$.

We have not found this to be a problem either in theory or practice, and Theorem 5 covers weak approximations.

Alternatively, one may just define successor states starting at $t = 1$ in (7.3.1). This only works well if rewards $r(s, a)$ depend on the state s but not the action a (e.g., in goal-oriented settings). If starting the definition at $t = 1$, m^π is an ordinary function provided the transition kernels $P(\mathrm{d}s'|s, a)$ of the environment are non-singular, ρ has positive density, and $\pi(\mathrm{d}a|s)$ is non-singular as well. Starting at $t = 1$ induces the following changes in the theorems:

- In the learning algorithm (7.5.2) for successor states, the term $\partial_\theta m^\pi_\theta(s, a, s, a)$ becomes $\gamma\, \partial_\theta m^\pi_\theta(s, a, s', a')$.
- The expression for the Q-function in Theorem 1 becomes $Q^\star(s, a) = r(s, a) + F(s, a, z_R)^\top z_R$, and likewise in Theorem 2. The $r(s, a)$ term covers the immediate reward at a state, since we have excluded $t = 0$ from the definition of successor states.
- In general the expression for optimal policies becomes

$$\pi_z(s) \triangleq \operatorname*{argmax}_a \{r(s, a) + F(s, a, z)^\top z\} \tag{7.8.37}$$

  which cannot be computed from z and F alone in the unsupervised training phase. The algorithm only makes sense for rewards that depend on s but not on a (e.g., in goal-oriented settings): then the policy π_z is equal to $\pi_z(s) \triangleq \operatorname{argmax}_a F(s, a, z)^\top z$ again.

7.9 Conclusion

The FB representation is a learnable mathematical object that "summarizes" a reward-free MDP. It provides near-optimal policies for any reward specified a posteriori, without planning. It is learned from black-box reward-free interactions with the environment. In practice, this unsupervised method performs comparably to goal-oriented methods for reaching arbitrary goals, but is also able to tackle more complex rewards in real time.

8 Conclusion

8.1 Summary of Contributions

The aim of this thesis was to provide theoretical solutions and insights to some fundamental issues in modern RL. This dissertation touched upon three major research areas: off-policy learning, exploration-exploitation dilemma in large state spaces, and self-supervision in RL.

- **Off-policy learning** is one of the main keys in scaling up RL agents as it enables the use of other streams of experiences than the one generated by themselves. In chapter 3, we showed how we can combine multi-step updates, for a rapid propagation of the errors, with function approximation in a provable way. We also provided a state-of-the-art finite sample analysis of gradient TD methods for policy evaluation. In chapter 4, we showed how to better exploit off-policy data to stabilize the policy improvement step. In particular, we proposed to learn in off-policy way the divergence between state-action visitations, which acts then as regularizer for the policy update. This improves upon conservative approaches which rely only on immediate action distributions for the regularization.

- Balancing between **exploration** and **exploitation** is critical for data-efficient learning i.e reaching optimal solution within a decent number of interactions. In this thesis, we focused on large state spaces. In chapter 5, we assumed that the optimal Q-function is smooth and we introduced ZoomRL, an efficient online algorithm that learns a data-driven partitioning and we proved that its regret scales with the covering dimension and does not depend on the number of states. In chapter 6, we assumed that both reward and transition kernel are linear and non-stationary. We introduced Opt-WLSVI, an efficient online algorithm that deploys a linear function approximation and uses discounting strategy to deal with the non-stationarity. We also fixed a serious flaw in the regret analysis of the non-stationary linear bandits, propagated in the literature.

- We focused in chapter 7 on **self-supervision** in reward-free environments. We proposed to learn representations, using interactions without reward signal, that encode a predictive occupancy map over states and actions. We

showed that we can read directly on these representation optimal policies for all reward functions.

A recurrent theme in the presented contributions is **generalisation**. In chapter 3, we devised convergent algorithms for policy evaluation that use off-policy multi-step traces while *generalizing over states via a linear function approximator*. In chapter 5, under smoothness assumptions, our proposed algorithm ZOOMRL *generalizes over large state-action spaces by learning a data-driven partitioning*. In chapter 6, our algorithm OPT-WLSVI *generalizes over states by using linear function approximation in a non-stationary linear environment*. Finally, in chapter 7 we focused on another kind of generalization which is the *generalization over reward functions*. Our learned forward-backward representations can return optimal behavior for any reward function or any goal state by computing a simple average.

8.2 FUTURE RESEARCH

To build reliable RL systems, there is far more to be done and I am eager to explore in the next few years some promising directions that I describe below.

8.2.1 Towards Fully Controllable Agents

Unsupervised learning provides a useful paradigm for the autonomous acquisition of task-agnostic knowledge. Our last-mentioned work in Chapter 7 is a step towards building agents that are fully controllable after first exploring their environment in an unsupervised way. While we tackled the representation learning problem, the exploration aspect remains a missing piece. The lack of exploration could limit the applicability of the learned representation to complex downstream tasks.

The open question that I would like to investigate is

Could we devise a principled unsupervised objective to discover behaviors or skills that are useful for both exploration (acquisition task-agnostic knowledge) and exploitation (solving a downstream task)?

8.2.2 Optimization for RL

While RL methods are traditionally dynamic-programming (DP) based algorithms, the field witnesses a recent trend that frames RL problems as well-posed optimization problems. Such algorithms have the benefit of avoiding inherent instability of DP and are more compatible with nonlinear/ linear function approximation. My first contribution in Chapter 3 lies in this direction. I would

like to investigate to what extent we can learn from established optimization tools to tackle RL problems.

To give a concrete example, I plan to look at the linear programming (LP) formulation of RL. This LP problem can be cast into a primal-dual problem where primal and dual variables are value functions and occupancy measures respectively. I would like to propose optimization-based algorithms tailored to this problem and study their convergence rate and sample complexity.

8.2.3 Statistical RL with General Function Approximation

It has been shown empirically that combining RL algorithms with neural network function approximators could lead to impressive performance on various tasks. However, the design of provable efficient RL algorithm with such general function classes remains elusive. In the past few years, there was some progress on this topic: Jiang et al. [2017] and Dong et al. [2020] introduce algorithms whose performance guarantees can be upper bounded in terms of the *Bellman rank*. However, these algorithms are not computationally efficient. Ayoub et al. [2020] and Wang et al. [2020] propose algorithms whose regret bounds scale with the *Eluder dimension*. However, we don't know if there are interesting function classes, beyond linear and generalized linear functions, with finite Eluder dimension.

In order to understand the role of function approximation in RL, the research question I would like to explore is

Could we develop better capacity measures of function class, adapted to the rich temporal structures of RL?

A Appendix from Chapter 3

A.1 Proof of Proposition 1

We compute $\mathbb{E}[A_k]$ and $\mathbb{E}[b_k]$ where expectation are over trajectories drawn by executing the behavior policy: $s_k, a_k, r_k, s_{k+1}, \dots s_t, a_t, r_t, s_{t+1} \dots$ where $s_k, a_k \sim d$, $r_t = r(s_t, a_t)$, $s_{t+1} \sim p(\cdot \mid s_t, a_t)$. We note that under stationarity of d, $\mathbb{E}[A_k] = E[A_0]$ and $\mathbb{E}[b_k] = \mathbb{E}[b_0]$. Let $\theta, \theta' \in \mathbb{R}^d$ and let $Q = \Phi\theta$ and $Q' = \Phi\theta'$ their respective Q-functions.

$$
\begin{aligned}
&\theta'^{\top}\mathbb{E}[A_k]\theta \\
&= \mathbb{E}\left[\sum_{t=0}^{\infty}(\lambda\gamma)^t\left(\prod_{i=1}^{t}\kappa_i\right)Q'(s_0,a_0)[\gamma\mathbb{E}_\pi Q(s_{t+1},.) - Q(s_t,a_t)]^\top\right] \\
&= \sum_{t=0}^{\infty}(\lambda\gamma)^t\mathbb{E}_{\substack{s_{0:t+1}\\a_{0:t}}}\left[Q'(s_0,a_0)\left(\prod_{i=1}^{t}\kappa_i\right)[\gamma\mathbb{E}_\pi Q(s_{t+1},.) - Q(s_t,a_t)]^\top\right] \\
&= \sum_{t=0}^{\infty}(\lambda\gamma)^t\mathbb{E}_{\substack{s_{0:t}\\a_{0:t}}}\left[Q'(s_0,a_0)\left(\prod_{i=1}^{t}\kappa_i\right)(\gamma\mathbb{E}_{s_{t+1}}[\mathbb{E}_\pi Q(s_{t+1},.)|s_t,a_t] - Q(s_t,a_t))\right] \\
&= \sum_{t=0}^{\infty}(\lambda\gamma)^t\mathbb{E}_{\substack{s_{0:t}\\a_{0:t}}}\left[Q'(s_0,a_0)\left(\prod_{i=1}^{t}\kappa_i\right)(\gamma\sum_{s'\in\mathcal{S}}\sum_{a'\in\mathcal{A}}p(s'\mid s_t,a_t)\pi(a'|s')Q(s',a') - Q(s_t,a_t))\right] \\
&= \sum_{t=0}^{\infty}(\lambda\gamma)^t\mathbb{E}_{\substack{s_{0:t}\\a_{0:t}}}\left[Q'(s_0,a_0)\left(\prod_{i=1}^{t}\kappa_i\right)(\gamma P^\pi Q(s_t,a_t) - Q(s_t,a_t))\right] \\
&= \sum_{t=0}^{\infty}(\lambda\gamma)^t\mathbb{E}_{\substack{s_{0:t-1}\\a_{0:t-1}}}\Big[Q'(s_0,a_0)\left(\prod_{i=1}^{t-1}\kappa_i\right) \\
&\quad\times\sum_{s'\in\mathcal{S}}\sum_{a'\in\mathcal{A}}p(s'\mid s_{t-1},a_{t-1})\kappa(a',s')\mu(a'|s')(\gamma P^\pi Q(s',a') - Q(s',a'))\Big] \\
&= \sum_{t=0}^{\infty}(\lambda\gamma)^t\mathbb{E}_{\substack{s_{0:t-1}\\a_{0:t-1}}}\left[Q'(s_0,a_0)\left(\prod_{i=1}^{t-1}\kappa_i\right)P^{\kappa\mu}(\gamma P^\pi - I)Q(s_{t-1},a_{t-1})\right] \\
&= \mathbb{E}_{s_0,a_0}\left[Q'(s_0,a_0)\sum_{t=0}^{\infty}(\lambda\gamma)^t(P^{\kappa\mu})^t(\gamma P^\pi - I)Q(x_0,a_0)\right]
\end{aligned}
$$

$$
\begin{aligned}
&= \mathbb{E}_{s_0,a_0}\left[Q'(s_0,a_0)(I-\lambda\gamma P^{\kappa\mu})^{-1}(\gamma P^{\pi}-I)Q(s_0,a_0)\right] \\
&= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \xi(s,a)Q'(s,a)(I-\lambda\gamma P^{\kappa\mu})^{-1}(\gamma P^{\pi}-I)Q(s,a) \\
&= Q'^{\top}\Xi(I-\lambda\gamma P^{\kappa\mu})^{-1}(\gamma P^{\pi}-I)Q
\end{aligned}
$$

So, $\theta'^{\top}\mathbb{E}[A_k]\theta = \theta'^{\top}\Phi^{\top}\Xi(I-\lambda\gamma P^{\kappa\mu})^{-1}(\gamma P^{\pi}-I)\Phi\theta \quad \forall\theta,\theta' \in \mathbb{R}^d$, which implies that:

$$\mathbb{E}[A_k] = \Phi^{\top}\Xi(I-\lambda\gamma P^{\kappa\mu})^{-1}(P^{\pi}-I)\Phi$$

$$
\begin{aligned}
&\theta^{\top}\mathbb{E}[b_k] \\
&= \mathbb{E}[\sum_{t=0}^{\infty}(\lambda\gamma)^t\left(\prod_{i=1}^{t}\kappa_i\right)r_tQ(s_0,a_0)] = \sum_{t=0}^{\infty}(\lambda\gamma)^t\mathbb{E}_{\substack{s_{0:t}\\a_{0:t}}}\left[Q(s_0,a_0)\left(\prod_{i=1}^{t}\kappa_i\right)r(s_t,a_t)\right] \\
&= \sum_{t=0}^{\infty}(\lambda\gamma)^t\mathbb{E}_{\substack{s_{0:t-1}\\a_{0:t-1}}}\left[Q(s_0,a_0)\left(\prod_{i=1}^{t-1}\kappa_i\right)\sum_{s'\in\mathcal{S}}\sum_{a'\in\mathcal{A}}p(s'\mid s_{t-1},a_{t-1})\kappa(a',s')\mu(a'\mid s')r(s',a')\right] \\
&= \sum_{t=0}^{\infty}(\lambda\gamma)^t\mathbb{E}_{\substack{s_{0:t-1}\\a_{0:t-1}}}\left[Q(s_0,a_0)\left(\prod_{i=1}^{t-1}\kappa_i\right)P^{\kappa\mu}r(s',a')\right] \\
&= \mathbb{E}_{s_0,a_0}\left[Q(s_0,a_0)(I-\lambda\gamma P^{\kappa\mu})^{-1}r(s_0,s_0)\right] \\
&= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\xi(s,a)Q(s,a)(I-\lambda\gamma P^{\kappa\mu})^{-1}r(s,a) = Q^{\top}\Xi(I-\lambda\gamma P^{\kappa\mu})^{-1}r
\end{aligned}
$$

So, $\theta^{\top}\mathbb{E}[b_k] = \theta^{\top}\Phi^{\top}\Xi(I-\lambda\gamma P^{\kappa\mu})^{-1}r \quad \forall\theta\in\mathbb{R}^d$, which implies that:

$$\mathbb{E}[b_k] = \Phi^{\top}\Xi(I-\lambda\gamma P^{\kappa\mu})^{-1}r$$

A.2 Proof of Proposition 2

$$
\begin{aligned}
&\textbf{MSPBE}(\theta) \\
&= \frac{1}{2}\|\Pi^{\mu}\mathcal{R}(\Phi\theta) - \Phi\theta||^2_{\Xi} \\
&= \frac{1}{2}\|\Pi^{\mu}\left(\mathcal{R}(\Phi\theta) - \Phi\theta\right)\|^2_{\Xi} \\
&= \frac{1}{2}\left(\Pi^{\mu}\left(\mathcal{R}(\Phi\theta) - \Phi\theta\right)\right)^{\top}\Xi\left(\Pi^{\mu}\left(\mathcal{R}(\Phi\theta) - \Phi\theta\right)\right) \\
&= \frac{1}{2}\left(\Phi^{\top}\Xi\left(\mathcal{R}(\Phi\theta) - \Phi\theta\right)\right)^{\top}(\Phi^{\top}\Xi\Phi)^{-1}\Phi^{\top}\Xi\left(\Phi(\Phi^{\top}\Xi\Phi)^{-1}\Phi^{\top}\Xi(\mathcal{R}(\Phi\theta) - \Phi\theta)\right) \\
&= \frac{1}{2}\|\Phi^{\top}\Xi\left(\mathbb{R}(\Phi\theta) - \Phi\theta\right)\|^2_{M^{-1}} \\
&= \frac{1}{2}\|\Phi^{\top}\Xi\left((I - \lambda\gamma P^{\mu\pi})^{-1}(\mathcal{T}^{\pi} - \lambda\gamma P^{\mu\pi})\Phi\theta - \Phi\theta\right)\|^2_{M^{-1}} \\
&= \frac{1}{2}\|\Phi^{\top}\Xi(I - \lambda\gamma P^{\mu\pi})^{-1}(\gamma P^{\pi} - I)\Phi\theta + \Phi^{\top}\Xi(I - \lambda\gamma P^{\mu\pi})^{-1}r\|^2_{M^{-1}} \\
&= \frac{1}{2}\|A\theta + b\|^2_{M^{-1}}
\end{aligned}
$$

A.3 Proof of Proposition 3

Let's show that $\mathbb{E}[\hat{A}_k] = A$. Let's Δ_t denotes $[\gamma\mathbb{E}_{\pi}\phi(s_{t+1},.)^{\top} - \phi(s_t,a_t)^{\top}]$

$$
\begin{aligned}
A &= \mathbb{E}\left[\sum_{t=k}^{\infty}(\lambda\gamma)^{t-k}\left(\prod_{i=k+1}^{t}\kappa_i\right)\phi(s_k,a_k)\Delta_t\right] \\
&= \mathbb{E}\left[\phi(s_k,a_k)\Delta_k + \sum_{t=k+1}^{\infty}(\lambda\gamma)^{t-k}\left(\prod_{i=k+1}^{t}\kappa_i\right)\phi(s_k,a_k)\Delta_t\right] \\
&= \mathbb{E}\left[\phi(s_k,a_k)\Delta_k + \sum_{t=k}^{\infty}(\lambda\gamma)^{t-k+1}\left(\prod_{i=k+1}^{t+1}\kappa_i\right)\phi(s_k,a_k)\Delta_{t+1}\right] \\
&= \mathbb{E}\Big[\phi(s_k,a_k)\Delta_k + \lambda\gamma\kappa(s_{k+1},a_{k+1})\phi(s_k,a_k)\Delta_{k+1} \\
&\quad + \sum_{t=k+1}^{\infty}(\lambda\gamma)^{t-k+1}\left(\prod_{i=k+1}^{t+1}\kappa_i\right)\phi(s_k,a_k)\Delta_{t+1}\Big]
\end{aligned}
$$

$$
\begin{aligned}
&\overset{(\star)}{=} \mathbb{E}\Big[\phi(s_k,a_k)\Delta_k + \lambda\gamma\kappa(s_k,a_k)\phi(s_{k-1},a_{k-1})\Delta_k \\
&\quad + \sum_{t=k+1}^{\infty}(\lambda\gamma)^{t-k+1}\left(\prod_{i=k+1}^{t+1}\kappa_i\right)\phi(s_k,a_k)\Delta_{t+1}\Big] \\
&= \mathbb{E}\Big[\Delta_k(\phi(s_k,a_k) + \lambda\gamma\kappa(s_k,a_k)\phi(s_{k-1},a_{k-1}) \\
&\quad + (\lambda\gamma)^2\kappa(s_k,a_k)\kappa(s_{k-1},a_{k-1})\phi(s_{k-2},a_{k-2}) + ...)\Big] \\
&= \mathbb{E}\left[\Delta_k\left(\sum_{i=0}^{k}(\lambda\gamma)^{k-i}\left(\prod_{j=i+1}^{k}\kappa_j\right)\phi(x_i,a_i)\right)\right] \\
&= \mathbb{E}[\Delta_k e_k] = \mathbb{E}[\hat{A}_k]
\end{aligned}
$$

we have used in the line $(\star)$ the fact that $\mathbb{E}[\kappa(s_{k+1},a_{k+1})\phi(s_k a_k)\Delta_{k+1}] = \mathbb{E}[\kappa(s_k,a_k)\phi(s_{k-1}a_{k-1})\Delta_k]$ thanks to the stationarity of the distribution d.
we have also denote by e_k the following vector:

$$
\begin{aligned}
e_k &= \sum_{i=0}^{k}(\lambda\gamma)^{k-i}\left(\prod_{j=i+1}^{k}\kappa_j\right)\phi(s_i,a_i) \\
&= \lambda\gamma\kappa_k\left(\sum_{i=0}^{k-1}(\lambda\gamma)^{k-1-i}\left(\prod_{j=i+1}^{k-1}\kappa_j\right)\phi(s_i,a_i)\right) + \phi(s_k,a_k) \\
&= \lambda\gamma\kappa_k e_{k-1} + \phi(s_k,a_k)
\end{aligned}
$$

Vector e_k corresponds to the eligibility traces defined in the proposition. Similarly, we could show that $\mathbb{E}_\mu[\hat{b}_k] = b$.

A.4 Convergence Rate Analysis

Let's recall the key quantities defined in the main article:

$$
\rho \triangleq \lambda_{\max}(A^\top M^{-1}A), \quad \delta \triangleq \lambda_{\min}(A^\top M^{-1}A), \quad L_G \triangleq \left\|\mathbb{E}\left[\hat{G}_k^\top \hat{G}_k \mid \mathcal{F}_k\right]\right\|
$$

We will make use of spectral properties of the matrix G provided in the appendix A of [Du et al., 2017]. it was shown that if we set $\beta = \frac{8\rho}{\lambda_{\min}(M)}$, the matrix G is diagonalizable with all its eigenvalues real and positive. It is a straightforward application of result from [Benzi and Simoncini, 2006]
Moreover, it was proved that G can be written as: $G = Q\Lambda Q^{-1}$ where Λ is a

diagonal matrix whose diagonal entries are the eigenvalues of G and Q consists of it eigenvectors as columns such that the condition number of Q is upper bounded by the one of M as follows:

$$c(Q)^2 \leq 8c(M)$$

Finally, the paper showed upper and lower bounds for the eigenvalues of G:

$$\lambda_{\max}(G) \leq 9c(M)\rho$$
$$\lambda_{\min}(G) \geq \frac{8}{9}\delta$$

Let's recall our updates:

$$z_{k+1} = z_k - \alpha_k(\hat{G}_k z_k - \hat{g}_k)$$

By subtracting $z^\star$ from both sides on the later equation and using the optimality condition $Gz^\star + g = 0$:

$$\Delta_{k+1} = \Delta_k - \alpha_k G\Delta_k + \alpha_k \left[Gz_k - g - (\hat{G}_k z_k - \hat{g}_k)\right] \quad \text{where} \quad \Delta_k \triangleq z_k - z^\star \tag{A.4.1}$$

By multiplying both sides by Q^{-1} and using the fact that $Q^{-1}G = \Lambda Q^{-1}$:

$$\begin{aligned} Q^{-1}\Delta_{k+1} &= Q^{-1}\Delta_k - \alpha_k Q^{-1}G\Delta_k + \alpha_k Q^{-1}\left[Gz_k - g - (\hat{G}_k z_k - \hat{g}_k)\right] \\ &= (I - \alpha_k\Lambda)Q^{-1}\Delta_k + \alpha_k Q^{-1}\left[Gz_k - g - (\hat{G}_k z_k - \hat{g}_k)\right] \end{aligned}$$

$$\begin{aligned} &\mathbb{E}\left[\left\|Q^{-1}\Delta_{k+1}\right\|^2 \mid \mathcal{F}_{k-1}\right] \\ &= \mathbb{E}\left[\left\|(I - \alpha_k)Q^{-1}\Delta_k + \alpha_k Q^{-1}\left[Gz_k - g - (\hat{G}_k z_k - \hat{g}_k)\right]\right\|^2 \mid \mathcal{F}_{k-1}\right] \\ &= \mathbb{E}\left[\left\|(I - \alpha_k\Lambda)Q^{-1}\Delta_k\right\|^2 \mid \mathcal{F}_{k-1}\right] \\ &\quad + 2\mathbb{E}\left[\left\langle (I - \alpha_k)Q^{-1}\Delta_k, \alpha_k Q^{-1}\left[Gz_k - g - (\hat{G}_k z_k - \hat{g}_k)\right]\right\rangle \mid \mathcal{F}_{k-1}\right] \\ &\quad + \alpha_k^2\mathbb{E}\left[\left\|Q^{-1}\left[Gz_k - g - (\hat{G}_k z_k - \hat{g}_k)\right]\right\|^2 \mid \mathcal{F}_{k-1}\right] \\ &= \left\|(I - \alpha_k\Lambda)Q^{-1}\Delta_k\right\|^2 + \alpha_k^2\mathbb{E}\left[\left\|Q^{-1}\left[Gz_k - g - (\hat{G}_k z_k - \hat{g}_k)\right]\right\|^2 \mid \mathcal{F}_{k-1}\right] \\ &\leq \left\|I - \alpha_k\Lambda\right\|^2\left\|Q^{-1}\Delta_k\right\|^2 + \alpha_k^2\mathbb{E}\left[\|Q^{-1}(\hat{G}_k z_k - \hat{g}_k)\|^2 \mid \mathcal{F}_{k-1}\right] \\ &= \left\|I - \alpha_k\Lambda\right\|^2\left\|Q^{-1}\Delta_k\right\|^2 + \alpha_k^2\mathbb{E}\left[\left\|Q^{-1}(\hat{G}_k\Delta_k + \hat{G}_k z^\star - \hat{g}_k)\right\|^2 \mid \mathcal{F}_{k-1}\right] \\ &\leq \left\|I - \alpha_k\Lambda\right\|^2\left\|Q^{-1}\Delta_k\right\|^2 + 2\alpha_k^2\mathbb{E}\left[\left\|Q^{-1}\hat{G}_k\Delta_k\right\|^2 \mid \mathcal{F}_{k-1}\right] + 2\alpha_k^2\mathbb{E}\left[\left\|Q^{-1}(\hat{G}_k z^\star + \hat{g}_k)\right\|^2 \mid \mathcal{F}_{k-1}\right] \end{aligned}$$

we use in the third line the fact that $\mathbb{E}\left[\hat{G}_k \mid \mathcal{F}_{k-1}\right] = G$ and $\mathbb{E}\left[\hat{g}_{k-1} \mid \mathcal{F}_{k-1}\right] = g$.

$$\begin{aligned}
\|I - \alpha_k \Lambda\|^2 &= \max\{|1 - \alpha_k \lambda_{\min}(G)|^2, |1 - \alpha_k \lambda_{\max}(G)|^2\} \\
&\leq 1 - 2\alpha_k \lambda_{\min} + \alpha_k^2 \lambda_{\max}^2 \\
&\leq 1 - 2\alpha_k \frac{8}{9}\delta + \alpha_k^2 9^2 c(M)^2 \rho^2 \\
&\leq 1 - 2\alpha_k \delta' + \alpha_k^2 9^2 c(M)^2 \rho^2 \quad \text{where} \quad \delta' \triangleq \frac{8}{9}\delta
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}\left[\left\|Q^{-1}\hat{G}_k\Delta_k\right\|^2 \mid \mathcal{F}_{k-1}\right] &\leq \|Q^{-1}\|^2 \mathbb{E}\left[\left\|\hat{G}_k\Delta_k\right\|^2 \mid \mathcal{F}_{k-1}\right] \\
&= \|Q^{-1}\|^2 \mathbb{E}\left[\Delta_k^\top \hat{G}_k^\top \hat{G}_k \Delta_k \mid \mathcal{F}_{k-1}\right] \\
&= \|Q^{-1}\|^2 \Delta_k^\top \mathbb{E}\left[\hat{G}_k^\top \hat{G}_k \mid \mathcal{F}_{k-1}\right] \Delta_k \\
&\leq \|Q^{-1}\|^2 \left\|\mathbb{E}\left[\hat{G}_k^\top \hat{G}_k \mid \mathcal{F}_k\right]\right\|^2 \Delta_k^\top \Delta_k \\
&\leq \|Q^{-1}\|^2 L_G \|\Delta_k\|^2 \\
&= \|Q^{-1}\|^2 L_G \|QQ^{-1}\Delta_k\|^2 \\
&\leq \|Q^{-1}\|^2 \|Q\|^2 L_G \|Q^{-1}\Delta_k\|^2 \\
&\leq c(Q)^2 L_G \|Q^{-1}\Delta_k\|^2
\end{aligned}$$

So, we have:

$$\begin{aligned}
&\mathbb{E}\left[\|Q^{-1}\Delta_{k+1}\|^2\right] \\
&\leq (1 - 2\alpha_k\delta' + \alpha_k^2 9^2 c(M)^2\rho^2 + 16\alpha_k^2 c(M) L_G)\mathbb{E}\left[\|Q^{-1}\Delta_k\|^2\right] + 2\alpha_k^2 \|Q^{-1}\|^2 \mathbb{E}\left[\|\hat{G}_k z^\star - \hat{g}_k)\|^2\right]
\end{aligned}$$

By selecting $\alpha_k = \frac{2\delta'}{\delta'^2(k+2) + 2\times 9^2 c(M)^2\rho^2 + 32 c(M) L_G} = \frac{2\delta'}{\delta'^2(k+2)+\zeta}$ with $\zeta = 2\times$

$9^2c(M)^2\rho^2 + 32c(M)L_G$, we get:

$$
\begin{aligned}
&\mathbb{E}\left[\|Q^{-1}\Delta_{k+1}\|^2\right] \\
&\leq (1-\delta'\alpha_k)\mathbb{E}\left[\|Q^{-1}\Delta_k\|^2\right] + 2\alpha_k^2\|Q^{-1}\|^2\mathbb{E}\left[\|\hat{G}_k z^\star - \hat{g}_k\|^2\right] \\
&= \frac{\delta'^2 k + \zeta}{\delta'^2(k+2)+\zeta}\mathbb{E}\left[\|Q^{-1}\Delta_k\|^2\right] + \frac{8\delta'^2}{(\delta'^2(k+2)+\zeta)^2}\|Q^{-1}\|^2\mathbb{E}\left[\|\hat{G}_k z^\star + \hat{g}_k)\|^2\right] \\
&\leq \left(\prod_{i=0}^{k}\frac{\delta'^2 i+\zeta}{\delta'^2(i+2)+\zeta}\right)\mathbb{E}\left[\|Q^{-1}\Delta_0\|^2\right] \\
&\quad + 8\delta'^2\sum_{i=0}^{k}\left(\prod_{j=i}^{k}\frac{\delta'^2 j+\zeta}{\delta'^2(j+2)+\zeta}\right)\frac{1}{(\delta'^2(i+2)+\zeta)^2}\|Q^{-1}\|^2\mathbb{E}\left[\|\hat{G}_i z^\star + \hat{g}_i)\|^2\right] \\
&= \frac{\zeta(\delta'^2+\zeta)}{(\delta'^2(k+1)+\zeta)((\delta'^2(k+2)+\zeta)}\mathbb{E}\left[\|Q^{-1}\Delta_0\|^2\right] \\
&\quad + 8\delta'^2\sum_{i=0}^{k}\frac{(\delta'^2(i+1)+\zeta)(\delta'^2 i+\zeta)}{(\delta'^2(k+1)+\zeta)((\delta'^2(k+2)+\zeta)}\frac{1}{(\delta'^2(i+2)+\zeta)^2}\|Q^{-1}\|^2\mathbb{E}\left[\|\hat{G}_i z^\star - \hat{g}_i\|^2\right] \\
&\leq \frac{\zeta(\delta'^2+\zeta)}{(\delta'^2(k+1)+\zeta)(\delta'^2(k+2)+\zeta)}\mathbb{E}\left[\|Q^{-1}\Delta_0\|^2\right] \\
&\quad + 8\delta'^2\sum_{i=0}^{k}\frac{1}{(\delta'^2(k+1)+\zeta)(\delta'^2(k+2)+\zeta)}\|Q^{-1}\|^2\mathbb{E}\left[\|\hat{G}_i z^\star - \hat{g}_i\|^2\right] \\
&\leq \frac{(\delta'+\zeta)^2}{(\delta'^2(k+1)+\zeta)^2}\mathbb{E}\left[\|Q^{-1}\Delta_0\|^2\right] \\
&\quad + 8\frac{\delta'^2(k+1)}{(\delta'^2(k+1)+\zeta)(\delta'^2(k+2)+\zeta)}\|Q^{-1}\|^2\sup_{i=0\ldots k}\mathbb{E}\left[\|\hat{G}_i z^\star + \hat{g}_i)\|^2\right] \\
&\leq \frac{(\delta'+\zeta)^2}{(\delta'^2(k+1)+\zeta)^2}\mathbb{E}\left[\|Q^{-1}\Delta_0\|^2\right] + \frac{8}{(\delta'^2(k+1)+\zeta)}\|Q^{-1}\|^2\sup_{i=0\ldots k}\mathbb{E}\left[\|\hat{G}_i z^\star - \hat{g}_i\|^2\right] \\
&\leq \frac{(\delta'+\zeta)^2\|Q^{-1}\|^2}{(\delta'^2(k+1)+\zeta)^2}\mathbb{E}\left[\|\Delta_0\|^2\right] + \frac{8\sigma^2\|Q^{-1}\|^2}{(\delta'^2(k+1)+\zeta)}(1+\|z^\star\|^2)
\end{aligned}
$$

Moreover, we have $\mathbb{E}\left[\|\Delta_{k+1}\|^2\right] = \mathbb{E}\left[\|QQ^{-1}\Delta_{k+1}\|^2\right] \leq \|Q\|^2\mathbb{E}\left[\|Q^{-1}\Delta_{k+1}\|^2\right]$.

Then, we get:

$$\begin{aligned}
\mathbb{E}\left[\|\Delta_{k+1}\|^2\right] &\leq \frac{(\delta'+\zeta)^2 c(Q)^2}{(\delta'^2(k+1)+\zeta)^2}\mathbb{E}\left[\|\Delta_0\|^2\right] + \frac{8\sigma^2 c(Q)^2}{(\delta'^2(k+1)+\zeta)}(1+\|z^\star\|^2) \\
&\leq \frac{8(\delta'+\zeta)^2 c(M)}{(\delta'^2(k+1)+\zeta)^2}\mathbb{E}\left[\|\Delta_0\|^2\right] + \frac{8^2\sigma^2 c(M)}{(\delta'^2(k+1)+\zeta)}(1+\|z^\star\|^2) \\
&= \frac{89^2(8\delta+9\zeta)^2 c(M)}{(8^2\delta^2(k+1)+9^2\zeta)^2}\mathbb{E}\left[\|\Delta_0\|^2\right] + \frac{9^2\times 8^2\sigma^2 c(M)}{(8^2\delta^2(k+1)+9^2\zeta)}(1+\|z^\star\|^2) \\
&= 9^2\times 8c(M)\left\{\frac{(8\delta+9\zeta)^2\mathbb{E}\left[\|\Delta_0\|^2\right]}{(8^2\delta^2(k+1)+9^2\zeta)^2} + \frac{8\sigma^2(1+\|z^\star\|^2)}{(8^2\delta^2(k+1)+9^2\zeta)}\right\}
\end{aligned}$$

The overall convergence rate is then equal to $O(1/k)$.

A.5 True on-line equivalence

In van Hasselt et al. [2014], the authors derived a true on-line update for GTD(λ) that empirically performed better than GTD(λ) with eligibility traces. Based on this work, we derive true on-line updates for our algorithm. The gradient off-policy algorithm was derived by turning the expected forward view into an expected backward view which can be sampled. In order to derive a true on-line update, we sample instead the forward view and then we turn the sampled forward view to an exact backward view using Theorem 1 in van Hasselt et al. [2014]. If k denotes the time horizon, we consider the sampled truncated interim forward return:

$$\forall t < k, \quad Y_t^k = \sum_{i=t}^{k-1}(\lambda\gamma)^{i-t}\left(\prod_{j=t+1}^{i}\kappa_j\right)\delta_i$$

where $\delta_i = r_i + \theta_t^\top \mathbb{E}_\pi \phi(s_{t+1},\cdot) - \theta_t^\top\phi(s_t,a_t)$, which gives us the sampled forward update of ω:

$$\forall k < t, \quad \omega_{t+1}^k = \omega_t^k + \alpha_t(Y_t^k - \phi(x_t,a_t)^\top\omega_t^k)\phi(x_t,a_t) \tag{A.5.1}$$

Proposition 9. *For any k, the parameter ω_k^k defined by the forward view (A.5.1) is equal to ω_k defined by the following backward view:*

$$\begin{aligned}
e_{-1}^\omega &= 0, \quad \forall k \geq 0 \\
e_k^\omega &= \lambda\gamma\kappa_k e_{k-1}^\omega + \alpha_k(1-\lambda\gamma\kappa_k\phi(s_k,a_k)^\top e_{k-1}^\omega)\phi(s_k,a_k) \\
\omega_{k+1} &= \omega_k + \delta_k e_k^\omega - \alpha_t\phi(s_k,a_k)^\top\omega_k\phi(s_k,a_k)
\end{aligned}$$

Proof. The return's temporal difference $Y_t^{k+1} - Y_t^k$ are related through:

$$\begin{aligned}
\forall t < k, \quad Y_t^{k+1} - Y_t^k &= \sum_{i=t}^{k} (\lambda\gamma)^{i-t} (\prod_{j=t+1}^{i} \kappa_j) \delta_i - \sum_{i=t}^{k-1} (\lambda\gamma)^{i-t} (\prod_{j=t+1}^{i} \kappa_j) \delta_i \\
&= (\lambda\gamma)^{k-t} \left(\prod_{j=t+1}^{k} \kappa_j \right) \delta_k \\
&= \lambda\gamma\kappa_{k+1} \left((\lambda\gamma)^{k-(t+1)} \left(\prod_{j=t+2}^{k} \kappa_j \right) \delta_k \right) \\
&= \lambda\gamma\kappa_{k+1} \left(Y_{t+1}^{k+1} - Y_{t+1}^{k} \right)
\end{aligned}$$

We could then apply Theorem 1 of van Hasselt et al. [2014] that give us the following backward view:

$$\begin{aligned}
e_0 &= \alpha_0 \phi(x_0, a_0) \\
e_t &= \lambda\gamma\kappa_t e_{t-1} + \alpha_t (1 - \lambda\gamma\kappa_k \phi(s_t, a_t)^\top e_{t-1}) \phi(s_t, a_t) \quad \forall t > 0 \\
\omega_{t+1} &= \omega_t + (Y_t^{t+1} - Y_t^t) e_t + \alpha_t (Y_t^t - \phi(s_t, a_t)^\top \omega_t) \phi(s_t, a_t) \\
&\overset{(\star)}{=} \omega_t + \delta_t e_t - \alpha_t \phi(s_t, a_t)^\top \omega_t \phi(s_t, a_t)
\end{aligned}$$

We used in the line $(\star)$ that $Y_t^{t+1} = \delta_t$ and $Y_t^t = 0$ □

The resulting detailed procedure is provided in Algorithm 8.

Note that when λ is equal to zero, the Algorithm 1 and 2 both reduce to the same update:

$$\begin{aligned}
\omega_{k+1} &= \omega_k + \alpha_k (\delta_k - \phi(s_k, a_k)^\top \omega_k) \phi(s_k, a_k) \\
\theta_{k+1} &= \theta_k - \alpha_k \phi(s_k, a_k)^\top w_k (\gamma \mathbb{E}_\pi[\phi(s_{k+1}, .)] - \phi(s_k, a_k)])
\end{aligned}$$

Algorithm 8 Gradient Off-policy with eligibility/Dutch traces

Given: target policy π, behavior policy μ

Initialize θ_0 and ω_0
for n = 0 ... **do**
  set $e^{\theta}_{-1} = e^{\omega}_{-1} = 0$
  for k = 0 ... end of episode **do**
    Observe s_k, a_k, r_k, s_{k+1} according to μ
    /* Update traces
    $e_k = \lambda\gamma\kappa(s_k, a_k)e_{k-1} + \phi(s_k, a_k)$
    /* Update Dutch traces
    $e^{\omega}_k = \lambda\gamma\kappa_k e^{\omega}_{k-1} + \alpha_k \left(1 - \lambda\gamma\kappa_k \phi(s_k, a_k)^\top e^{\omega}_{k-1}\right) \phi(s_k, a_k)$
    /* Update parameters
    $\delta_k = r_k + \gamma\theta_k^\top \mathbb{E}_\pi \phi(s_{k+1}, .) - \theta_k^\top \phi(s_k, a_k)$
    $\omega_{k+1} = \omega_k + \delta_k e^{\omega}_k - \alpha_k \phi(s_k, a_k)^\top \omega_k \phi(s_k, a_k)$
    $\theta_{k+1} = \theta_k - \alpha_k \omega_k^\top e_k \left(\gamma \mathbb{E}_\pi[\phi(s_{k+1}, .)] - \phi(s_k, a_k)\right)$
  end for
end for

B Appendix For Chapter 4

B.1 Proof of Lemma 2

According to performance difference lemma 1, we have

$$
\begin{aligned}
J(\pi') &= J(\pi) + \mathbb{E}_{s\sim d_\rho^{\pi'}} \mathbb{E}_{a\sim\pi'(\cdot|s)} \left[A^\pi(s,a)\right] \\
&= J(\pi) + \left(\mathbb{E}_{s\sim d_\rho^{\pi}} \mathbb{E}_{a\sim\pi'(\cdot|s)} \left[A^\pi(s,a)\right] + \int_{s\in\mathcal{S}} \mathbb{E}_{a\sim\pi'(\cdot|s)} \left[A^\pi(s,a)\right] (d_\rho^{\pi'}(s) - d_\rho^{\pi}(s)) ds \right) \\
&\geq J(\pi) + \left(\mathbb{E}_{s\sim d_\rho^{\pi}} \mathbb{E}_{a\sim\pi'(\cdot|s)} \left[A^\pi(s,a)\right] - \int_{s\in\mathcal{S}} |\mathbb{E}_{a\sim\pi(\cdot|s)} \left[A^\pi(s,a)\right]| \cdot |d_\rho^{\pi'}(s) - d_\rho^{\pi}(s)| ds \right) \\
&\geq J(\pi) + \left(\mathbb{E}_{s\sim d_\rho^{\pi}} \mathbb{E}_{a\sim\pi'(\cdot|s)} \left[A^\pi(s,a)\right] - \epsilon^\pi \int_{s\in\mathcal{S}} |d_\rho^{\pi'}(s) - d_\rho^{\pi}(s)| ds \right) \\
&\geq J(\pi) + \left(\mathbb{E}_{s\sim d_\rho^{\pi}} \mathbb{E}_{a\sim\pi'(\cdot|s)} \left[A^\pi(s,a)\right] - \epsilon^\pi D_{\mathrm{TV}}(d_\rho^{\pi'} \| d_\rho^{\pi}) \right) \\
&= L_\pi(\pi') - \epsilon^\pi D_{\mathrm{TV}}(d_\rho^{\pi'} \| d_\rho^{\pi})
\end{aligned}
$$

where $\epsilon^\pi = \max_s |\mathbb{E}_{a\sim\pi'(\cdot|s)} \left[A^\pi(s,a)\right]|$ and D_{TV} is total variation distance. The first inequality follows from Cauchy-Schwartz inequality.

B.2 Score Function Estimator of the Gradient with Respect to the Policy

$$
\begin{aligned}
\nabla_{\pi'} \mathbb{E}_{\substack{s\sim\rho \\ a\sim\pi'}} [g(s,a)] &= \nabla_{\pi'} \int g(s,a)\rho(s)\pi'(a \mid s) \\
&= \int g(s,a)\rho(s)\nabla_{\pi'}\pi'(a \mid s) \\
&= \mathbb{E}_{\substack{s\sim\rho \\ a\sim\pi'}} [g(s,a)\nabla_{\pi'} \log \pi'(a \mid s)]
\end{aligned}
$$

$$
\begin{aligned}
&\nabla_{\pi'} \mathbb{E}_{(s,a)\sim\mu_\rho^{\pi_i}} \left[\phi^\star \left((g - \gamma P^{\pi'} g)(s,a) \right) \right] \\
&= \mathbb{E}_{(s,a)\sim\mu_\rho^{\pi_i}} \left[\nabla_{\pi'} \phi^\star \left((g - \gamma P^{\pi'} g)(s,a) \right) \right] \\
&= \mathbb{E}_{(s,a)\sim\mu_\rho^{\pi_i}} \left[\frac{\partial \phi^\star}{\partial t} \left((g - \gamma P^{\pi'} g)(s,a) \right) \nabla_{\pi'} (g - \gamma \mathbb{P}^{\pi'} g) \right] \\
&= -\gamma \mathbb{E}_{(s,a)\sim\mu_\rho^{\pi_i}} \left[\frac{\partial \phi^\star}{\partial t} \left((g - \gamma P^{\pi'} g)(s,a) \right) \nabla_{\pi'} \int g(s',a') P(s' \mid s,a) \pi'(a' \mid s')) \right] \\
&= -\gamma \mathbb{E}_{(s,a)\sim\mu_\rho^{\pi_i}} \left[\frac{\partial \phi^\star}{\partial t} \left((g - \gamma P^{\pi'} g)(s,a) \right) \mathbb{E}_{\substack{s'\sim P(\cdot|s,a) \\ a'\sim\pi'(\cdot|s')}} \left[g(s',a') \nabla_{\pi'} \log \pi'(a' \mid s') \right] \right]
\end{aligned}
$$

B.3 Comparison with AlgaeDICE

Both the recent AlgaeDICE [Nachum et al., 2019b] and our present work propose regularisation based on discounted state-action visitation distribution but in different ways. Firstly, AlgaeDICE is initially designed to find an optimal policy given a batch of training data. They alter the objective function itself i.e the policy performance $J(\pi)$ by adding the divergence between the discounted state-action visitation distribution and training distribution, while our approach adds the divergence term to $L_\pi(\pi')$. The latter is a first order Taylor approximation of the policy performance $J(\pi')$. Therefore, our approach could be seen as a mirror descent that uses the divergence as a proximity term. Secondly, their training objective is completely different from ours. Their method ends up being an off-policy version of the actor-critic method.

We implemented the AlgaeDICE min-max objective to replace our surrogate min-max objective in the PPO training procedure i.e at each iteration, we sample rollouts from the current policy and update the actor and the critic of AlgaeDICE for 10 epochs. Empirically, we observed that AlgaeDICE objective is very slow to train in this setting. This was expected as it is agnostic to training data while our method leverages the fact that the data is produced by the current policy and estimates advantage using on-policy multi-step Monte Carlo. So our approach is more suitable than AlgaeDICE in this setting. However, AlgaeDICE, as an off-policy method, would be better when storing all history of transitions and updating both actor and critic after each transition, as shown in Nachum et al. [2019b].

B.4 Additional Empirical Results on MuJoCo

See Figure B.1

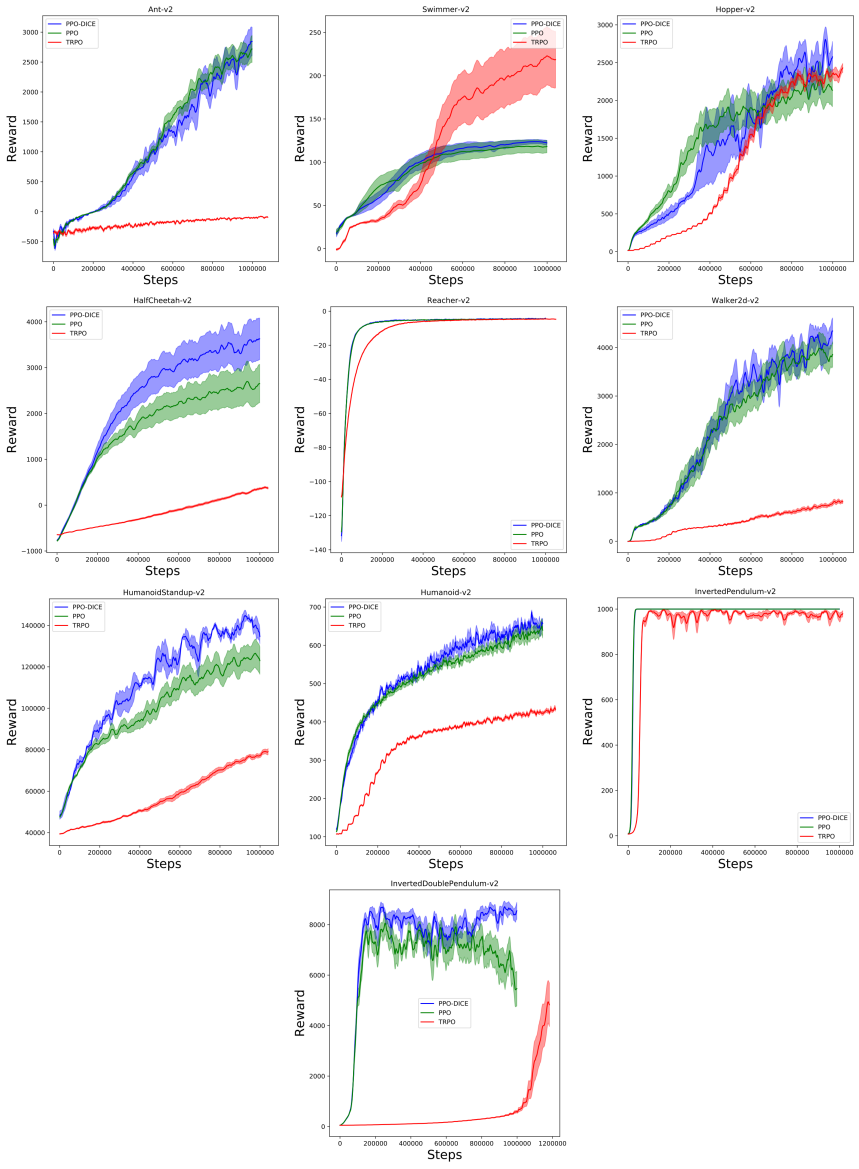

Figure B.1: Our method with KL divergences in comparison to PPO and TRPO on MuJoCo, with 10 seeds. Standard error shaded.

B.5 HYPERPARAMETERS

For the OpenAI Gym environments we use the default hyperparameters in Kostrikov [2018].

Parameter name	MuJoCo	Atari
Number of minibatches	4	4
Discount γ	0.99	0.99
Optimizer	Adam	Adam
Learning rate	3e-4	2.5e-4
PPO clip parameter	0.2	0.1
PPO epochs	10	4
Number of processes	1	8
GAE λ	0.95	0.95
Entropy coef	0	0.01
Value loss coef	0.5	0.5
Number of forward steps per update	2048	128

Table B.1: A complete overview of used hyper parameters for all methods.

C Appendix For Chapter 5

C.1 Omitted proofs for the Lipschitz setting

C.1.1 Proof of Lemma 3

(a) It is obvious that $\cup_{B\in\mathcal{B}_h^k} \texttt{dom}_h^k(B) \subset \cup_{B\in\mathcal{B}_h^k} B$. Let $x \in \cup_{B\in\mathcal{B}_h^k} B$. Consider a smallest radius ball B in $\mathcal{B}_h^k$ that contains x. Hence, $x \in \texttt{dom}_h^k(B)$. This shows that $\cup_{B\in\mathcal{B}_h^k} B \subset \cup_{B\in\mathcal{B}_h^k} \texttt{dom}_h^k(B)$ and consequently $\cup_{B\in\mathcal{B}_h^k} B = \cup_{B\in\mathcal{B}_h^k} \texttt{dom}_h^k(B)$. Moreover, $\cup_{B\in\mathcal{B}_h^k} B = \mathcal{S}\times\mathcal{A}$ as it contains the initial ball which covers the whole space.

(b) Let $(B, B') \in \mathcal{B}_h^k$ two balls of radius $r > 0$. Without loss of generality, we suppose that B is created in episode $\tau \leq k$ with parent ball $B^{\texttt{pa}}$ and B' is created before τ. According to the activation step in ZoomRL algorithm, (s_h^τ, a_h^τ) is the center of B and $(s_h^\tau, a_h^\tau) \in \texttt{dom}_h^\tau(B^{\texttt{pa}})$. By the definition of a ball's domain, $(s_h^\tau, a_h^\tau) \notin B'$, which proves that $\texttt{dist}(B, B') > r$.

C.1.2 Proof of Lemma 4

Proof. We fix $B \in \mathcal{B}_h^k$. For notation simplicity, denote $t = n_h^k(B)$. We have:

$$\begin{aligned}
\widehat{\mathrm{Q}}_h^k(B) &= (1-\alpha_t)\cdot \widehat{\mathrm{Q}}_h^{k_t}(B) + \alpha_t \cdot \left(r_h(x_h^{k_t}, a_h^{k_t}) + \widehat{\mathrm{V}}_{h+1}^{k_t}(x_{h+1}^{k_t}) + u_t + 2L\cdot \mathtt{rad}(B)\right) \\
&= (1-\alpha_t)\cdot \Big((1-\alpha_{t-1})\cdot \widehat{\mathrm{Q}}_h^{k_{t-1}}(B) \\
&\quad + \alpha_{t-1}\cdot \left(r_h(x_h^{k_{t-1}}, a_h^{k_{t-1}}) + \widehat{\mathrm{V}}_{h+1}^{k_{t-1}}(x_{h+1}^{k_{t-1}}) + u_{t-1} + 2L\cdot \mathtt{rad}(B)\right)\Big) \\
&\quad + \alpha_t \cdot \left(r_h(x_h^{k_t}, a_h^{k_t}) + \widehat{\mathrm{V}}_{h+1}^{k_t}(x_{h+1}^{k_t}) + u_i + 2L\cdot \mathtt{rad}(B)\right) \\
&= \dots \\
&= \prod_{i=1}^{t}(1-\alpha_i)H + \sum_{i=1}^{t}\alpha_i \prod_{j=i+1}^{t}(1-\alpha_j)\left(r_h(x_h^{k_i}, a_h^{k_i}) + \widehat{\mathrm{V}}_{h+1}^{k_i}(x_{h+1}^{k_i}) + u_i + 2L\cdot \mathtt{rad}(B)\right) \\
&= \alpha_t^0 \cdot H + \sum_{i=1}^{t}\alpha_t^i \cdot \left(r_h(x_h^{k_i}, a_h^{k_i}) + \widehat{\mathrm{V}}_{h+1}^{k_i}(x_{h+1}^{k_i}) + u_i + 2L\cdot \mathtt{rad}(B)\right),
\end{aligned}$$

where the first step follows from the update rule of $\widehat{\mathrm{Q}}_h^k$, the second step follows from the update rule for $\widehat{\mathrm{Q}}_h^{k_{t-1}}$, the third step follows from recursively representing $\widehat{\mathrm{Q}}_h^{k_i}$ using $\widehat{\mathrm{Q}}_h^{k_{i-1}}$ until $i = 1$, and the last step follows from the definition of α_t^0 and α_t^i. □

C.1.3 Proof of Lemma 5

Proof. Let $B \in \mathcal{B}_h^k$ and $(s,a) \in \mathtt{dom}_h^k(B)$.

Since $\sum_{i=0}^t \alpha_t^i = 1$, we have that $Q_h^\star(s,a) = \alpha_t^0 Q_h^\star(s,a) + \sum_{i=1}^t \alpha_t^i Q_h^*(s,a)$.

By the Lipschitz assumption 4 and the fact $\forall i \in [t], (s_h^{k_i}, a_h^{k_i}) \in B$ and $(s,a) \in B$, we have:

$$|Q_h^\star(s_h^{k_i}, a_h^{k_i}) - Q_h^\star(s,a)| \le L\cdot \mathtt{dist}((x_h^{k_i}, a_h^{k_i}), (x,a)) \le 2L\cdot \mathtt{rad}(B).$$

Then we have

$$Q_h^\star(s,a) \ge \alpha_t^0 Q_h^\star(s,a) + \sum_{i=1}^{t}\alpha_t^i \left(Q_h^\star(s_h^{k_i}, a_h^{k_i}) - 2L\cdot \mathtt{rad}(B)\right) \tag{C.1.1}$$

$$Q_h^\star(s,a) \le \alpha_t^0 Q_h^\star(s,a) + \sum_{i=1}^{t}\alpha_t^i \left(Q_h^\star(s_h^{k_i}, a_h^{k_i}) + 2L\cdot \mathtt{rad}(B)\right). \tag{C.1.2}$$

By Bellman equation, we have $Q_h^\star(s_h^{k_i}, a_h^{k_i}) = r_h(s_h^{k_i}, a_h^{k_i}) + [P_h V_{h+1}^\star](s_h^{k_i}, a_h^{k_i})$. Recall $[\widehat{P}_h^{k_i} V_{h+1}](s_h^{k_i}, a_h^{k_i}) = V_{h+1}(s_{h+1}^{k_i})$, we have:

$$Q_h^\star(s_h^{k_i}, a_h^{k_i}) = r_h(s_h^{k_i}, a_h^{k_i}) + [(P_h - \widehat{P}_h^{k_i})V_{h+1}^\star](s_h^{k_i}, a_h^{k_i}) + V_{h+1}^\star(s_{h+1}^{k_i}).$$

Substitute the above equality into Eq. (C.1.1) and (C.1.2), we have:

$$Q_h^\star(s,a) \geq \alpha_t^0 Q_h^\star(x,a)) + \sum_{i=1}^{t} \alpha_t^i \Big(r_h(x_h^{k_i}, a_h^{k_i}) + [(P_h - \widehat{P}_h^{k_i}) V_{h+1}^\star](x_h^{k_i}, a_h^{k_i}) + V_{h+1}^\star(x_{h+1}^{k_i}) - 2L \cdot \mathtt{rad}(B) \Big) \tag{C.1.3}$$

and

$$Q_h^\star(s,a) \leq \alpha_t^0 Q_h^\star(s,a)) + \sum_{i=1}^{t} \alpha_t^i \Big(r_h(x_h^{k_i}, a_h^{k_i}) + [(P_h - \widehat{P}_h^{k_i}) V_{h+1}^\star](x_h^{k_i}, a_h^{k_i}) + V_{h+1}^\star(x_{h+1}^{k_i}) + 2L \cdot \mathtt{rad}(B) \Big) \tag{C.1.4}$$

Subtracting the formula in Lemma 4 from the two above inequalities, we have:

$$\widehat{\mathrm{Q}}_h^k(B) - Q_h^\star(s,a) \geq \sum_{i=1}^{t} \alpha_t^i \left((\widehat{\mathrm{V}}_{h+1}^{k_i} - V_{h+1}^\star)(x_{h+1}^{k_i}) + [(\widehat{P}_h^{k_i} - P_h) V_{h+1}^*](x_h^{k_i}, a_h^{k_i}) + u_i \right) \tag{C.1.5}$$

and

$$\widehat{\mathrm{Q}}_h^k(B) - Q_h^\star(s,a) \leq \alpha_t^0 H + \sum_{i=1}^{t} \alpha_t^i \Big((\widehat{\mathrm{V}}_{h+1}^{k_i} - V_{h+1}^\star)(x_{h+1}^{k_i}) + [(\widehat{P}_h^{k_i} - P_h) V_{h+1}^*](x_h^{k_i}, a_h^{k_i}) + u_i + 4L \cdot \mathtt{rad}(B) \Big) \tag{C.1.6}$$

High probability bounds on the sampling noise

To ensure that our estimates concentrate around the true optimal Q-values, we need to ensure that the noise terms $[(\widehat{P}_h^{k_i} - P_h) V_{h+1}^*](x_h^{k_i}, a_h^{k_i})$, due to the next states sampling, are not large.

For each ball $B \in \mathcal{B}_h^k$, k_i is the episode of which B was selected as step h for the i-th time. Let $\mathcal{F}_t$ be the σ-field generated by all the random variables until episode t, step h. As $\{k_i = t\} \in \mathcal{F}_t$, the random variable k_i is a stopping time. By definition for any $i \geq 0$, $k_i \leq k_{i+1}$ so the σ-algebra $\mathcal{F}_{k_i}$ at time k_i satisfies $\mathcal{F}_{k_i} \subset \mathcal{F}_{k_{i+1}}$ (see Lemma 17). Let's denote $\mathcal{G}_i = \mathcal{F}_{\tau_{i+1}}$. Then, $(\mathcal{G}_i)_i$ is a filtration. Moreover, via optional stopping [Chow and Teicher, 1998], $\left(\mathbb{I}(k_i \leq K)[(\widehat{P}_h^{k_i} - P) V_{h+1}^*](x_h^{k_i}, a_h^{k_i}) \right)_{i=1}^{\tau}$ is a $2H$-bounded martingale difference sequence w.r.t the filtration $(\mathcal{G}_i)_{i \geq 0}$. By Azuma-Hoeffding 15, we have $\forall t > 0, \tau \in [K]$

$$\Pr\left[\left| \sum_{i=1}^{\tau} \alpha_\tau^i \cdot \mathbb{I}(k_i \leq K) \cdot [(\widehat{P}_h^{k_i} - P_h) V_{h+1}^*](x_h^{k_i}, a_h^{k_i}) \right| \geq t \right] \leq 2 \exp\left(\frac{-t^2}{8H^2 \sum_{i=1}^{\tau} (\alpha_\tau^i)^2} \right)$$

Let $p \in (0,1)$, by setting $2\exp\left(\frac{-t^2}{8H^2\sum_{i=1}^{\tau}(\alpha_\tau^i)^2}\right) = \frac{p}{2HK^2}$, we have that for all $\tau \in [k]$ with probability at least $1 - \frac{p}{2HK^2}$:

$$\left|\sum_{i=1}^{\tau} \alpha_\tau^i \cdot \mathbb{I}(k_i \le K) \cdot [(\widehat{P}_h^{k_i} - P_h)V_{h+1}^*](x_h^{k_i}, a_h^{k_i})\right| \le 2\sqrt{2}H\sqrt{\sum_{i=1}^{\tau}(\alpha_\tau^i)^2 \ln(4HK^2/p)}$$
$$\le 4\sqrt{\frac{H^3 \ln(4HK^2/p)}{\tau}} = 4\sqrt{\frac{H^3\iota}{\tau}}$$

where the second inequality follows from $\sum_{i=1}^{\tau}(\alpha_\tau^i)^2 \le \frac{2H}{\tau}$ for any $\tau > 0$ (see lemma 16). Then by union bound over $\tau \in [K]$, we have with probability at least $1 - \frac{p}{2HK}$

$$\forall \tau \in [K], \quad \left|\sum_{i=1}^{\tau} \alpha_\tau^i \cdot \mathbb{I}(k_i \le K) \cdot [(\widehat{P}_h^{k_i} - P_h)V_{h+1}^*](x_h^{k_i}, a_h^{k_i})\right| \le 4\sqrt{\frac{H^3\iota}{\tau}}$$

Since the above inequality holds for all $\tau \in [K]$, it also holds for $\tau = t = n_h^h(B) \le K$. We also have that $\mathbf{I}(k_i \le K) = 1$ for any $i \le n_h^h(B)$. As $|\mathcal{B}_h^k| \le K$ for all $(h,k) \in [H] \times [K]$, using union bound for all balls and for all steps, we have with probability at least $1 - p/2$: $\forall (h,k) \in [H] \times [K]$ and for all ball $B \in \mathcal{B}_h^k$,

$$\left|\sum_{i=1}^{t} \alpha_t^i \cdot [(\widehat{P}_h^{k_i} - P_h)V_{h+1}^*](x_h^{k_i}, a_h^{k_i})\right| \le 4\sqrt{\frac{H^3\iota}{t}}, \text{where } t = n_h^k(B) \tag{C.1.7}$$

According to Lemma 16, we have $1/\sqrt{t} \le \sum_{i=1} \frac{\alpha_t^i}{t} \le 2/\sqrt{t}$. This implies

$$4\sqrt{\frac{H^3\iota}{t}} \le 4\sqrt{H^3\iota} \cdot \sum_{i=1}^{t} \frac{\alpha_t^i}{t} = \sum_{i=1}^{t} \alpha_t^i u_i = \beta_t/2 \le 8\sqrt{\frac{H^3\iota}{t}}$$

Then Eq (C.1.7) gives that, with probability at least $1 - p/2$: $\forall (h,k) \in [H] \times [K]$ and for all ball $B \in \mathcal{B}_h^k$,

$$\left|\sum_{i=1}^{t} \alpha_t^i \cdot [(\widehat{P}_h^{k_i} - P_h)V_{h+1}^*](x_h^{k_i}, a_h^{k_i})\right| \le \beta_t/2, \text{where } t = n_h^k(B) \tag{C.1.8}$$

Optimism of Q-values: Lemma 5 (a)

We proceed by induction. By definition, we have $\widehat{Q}_{H+1}^k = Q_{H+1}^* = 0$ which implies $Q_{H+1}^k(B) - Q_{H+1}^*(s,a) = 0$. Assume that $Q_{h+1}^k(B) - Q_{h+1}^*(s,a) \ge 0$.

Let $i \in [1,t]$, We have $V^\star_{h+1}(s^{k_i}_{h+1}) = Q^\star_{h+1}(s^{k_i}_{h+1}, \pi^\star_{h+1}(s^{k_i}_{h+1}))$. As the set of domains of active balls covers the entire space, there exists $B^\star \in \mathcal{B}^k_{h+1}$ such that $(s^{k_i}_{h+1}, \pi^\star_{h+1}(s^{k_i}_{h+1})) \in \texttt{dom}^k_{h+1}(B^*)$. By the definition of index, we have $\texttt{index}^k_{h+1}(B^\star) = L \cdot \texttt{rad}(B^\star) + \widehat{\mathrm{Q}}^k_{h+1}(\tilde{B}^\star) + L \cdot \texttt{dist}(\tilde{B}^\star, B^\star)$ for some active ball $\tilde{B}^\star$.

We have

$$
\begin{aligned}
&\widehat{\mathrm{V}}^k_{h+1}(s^{k_i}_{h+1}) - V^\star_{h+1}(s^{k_i}_{h+1}) \\
&= \min\{H, \max_{B \in \texttt{rel}^k_{h+1}(s^{k_i}_{h+1})} \texttt{index}^k_{h+1}(B)\} - Q^\star_{h+1}(s^{k_i}_{h+1}, \pi^\star(s^{k_i}_{h+1})) \\
&\geq \max_{B \in \texttt{rel}^k_{h+1}(s^{k_i}_{h+1})} \texttt{index}^k_{h+1}(B) - Q^\star_{h+1}(s^{k_i}_{h+1}, \pi^\star(s^{k_i}_{h+1})) \\
&\geq \texttt{index}^k_{h+1}(B^\star) - Q^\star_{h+1}(s^{k_i}_{h+1}, \pi^\star(s^{k_i}_{h+1})) \\
&= L \cdot \texttt{rad}(B^\star) + \widehat{\mathrm{Q}}^k_{h+1}(\tilde{B}^\star) + L \cdot \texttt{dist}(\tilde{B}^\star, B^\star) - Q^\star_{h+1}(s^{k_i}_{h+1}, \pi^\star(s^{k_i}_{h+1})) \\
&\geq L \cdot \texttt{rad}(B^\star) + Q^\star_{h+1}(s_{\tilde{B}^\star}, a_{\tilde{B}^\star}) + L \cdot \texttt{dist}(\tilde{B}^\star, B^\star) - Q^\star_{h+1}(s^{k_i}_{h+1}, \pi^\star(s^{k_i}_{h+1})) \\
&\geq L \cdot \texttt{rad}(B^\star) + Q^\star_{h+1}(s_{B^\star}, a_{B^\star}) - Q^\star_{h+1}(s^{k_i}_{h+1}, \pi^\star(s^{k_i}_{h+1})) \\
&\geq 0,
\end{aligned}
$$

where $(s_{B^\star}, a_{B^\star})$ and $(s_{\tilde{B}^\star}, a_{\tilde{B}^\star})$ denote respectively the centers of balls $B^\star$ and $\tilde{B}^\star$. The first inequality follows from $Q^\star_{h+1}(s,a) \leq H$ for any state-action pair (s,a). The third inequality follows from the induction hypothesis. The fourth and the last inequalities follow from Lipschitz assumption 4

Therefore, we have

$$
\begin{aligned}
Q^k_h(B) - Q^*_h(s,a) &\geq \sum_{i=1}^{t} \alpha^i_t \cdot \left((\widehat{\mathrm{V}}^{k_i}_{h+1} - V^\star_{h+1})(s^{k_i}_{h+1}) + [(\widehat{P}^{k_i}_h - P_h)V^\star_{h+1}](s^{k_i}_h, a^{k_i}_h) + u_i \right) \\
&\geq -\beta_t/2 + \beta_t/2 = 0.
\end{aligned}
$$

Upper bound: lemma 5 (b)

We have:

$$
\begin{aligned}
&\widehat{\mathrm{Q}}_h^k(B) - Q_h^\star(s,a) \\
&\leq \alpha_t^0 \cdot H + \sum_{i=1}^{t} \alpha_t^i \cdot \Big((V_{h+1}^{k_i} - V_{h+1}^*)(x_{h+1}^{k_i}) + [(\widehat{P}_h^{k_i} - P_h)V_{h+1}^*](x_h^{k_i}, a_h^{k_i}) + u_i + 4r(B) \Big). \\
&\leq \alpha_t^0 H + \sum_{i=1}^{t} \alpha_t^i \cdot (\widehat{\mathrm{V}}_{h+1}^{k_i} - V_{h+1}^*)(x_{h+1}^{k_i}) + \beta_t/2 + \sum_{i=1}^{t} \alpha_t^i u_i + 4L \sum_{i=1}^{t} \alpha_t^i \, \mathtt{rad}(B) \\
&\leq \alpha_t^0 H + \beta_t + 4L \cdot \mathtt{rad}(B) + \sum_{i=1}^{t} \alpha_t^i \cdot (\widehat{\mathrm{V}}_{h+1}^{k_i} - V_{h+1}^*)(x_{h+1}^{k_i}),
\end{aligned}
$$

where the second inequality follows from the inequality C.1.8. The third inequality follows from $\sum_{i=1}^{t} \alpha_t^i \leq 1$ □

C.1.4 Proof of lemma 7

Let B_h^k the ball selected at step h of episode k and $B_{\mathtt{init}}$ be the initial ball of radius one that covers the whole space. We need to distinguish between cases where $B_h^k = B_{\mathtt{init}}$ or not. By the selection step in ZOOMRL algorithm, we have $\max_{B \in \mathtt{rel}_h^k(s_h^k)} \mathtt{index}_h^k(B) = \mathtt{index}_h^k(B_h^k)$ and $\pi_k(s_h^k) = a_h^k$.

1. **Case of $B_h^k \neq B_{\mathtt{init}}$:** We denote $B_h^{k,\mathtt{pa}}$ the parent of B_h^k.

$$
\begin{aligned}
\delta_h^k &= (\widehat{\mathrm{V}}_h^k - V_h^{\pi_k})(s_h^k) \\
&\leq \max_{B \in \mathtt{rel}_h^k(s_h^k)} \mathtt{index}_h^k(B) - V_h^{\pi_k}(s_h^k) \\
&= \mathtt{index}_h^k(B_h^k) - Q_h^{\pi_k}(s_h^k, a_h^k) \\
&\leq L \cdot \mathtt{rad}(B_h^k) + \widehat{\mathrm{Q}}_h^k(B_h^{k,\mathtt{pa}}) + L \cdot \mathtt{dist}(B_h^{k,\mathtt{pa}}, B_h^k) - Q_h^{\pi_k}(s_h^k, a_h^k) \\
&\leq L \cdot \mathtt{rad}(B_h^k) + \widehat{\mathrm{Q}}_h^k(B_h^{k,\mathtt{pa}}) + L \cdot \mathtt{rad}(B_h^{k,\mathtt{pa}}) - Q_h^{\pi_k}(s_h^k, a_h^k) \\
&= 3L \cdot \mathtt{rad}(B_h^k) + \underbrace{\widehat{\mathrm{Q}}_h^k(B_h^{k,\mathtt{par}}) - Q_h^*(s_h^k, a_h^k)}_{q_1} + (Q_h^* - Q_h^{\pi_k})(s_h^k, a_h^k)
\end{aligned}
$$

The third inequality follows from the fact that the center of B_h^k is in $B_h^{k,\mathtt{pa}}$ and the last equality follows from $\mathtt{rad}(B_h^{k,\mathtt{pa}}) = \mathtt{rad}(B_h^k)$.

Since $(x_h^k, a_h^k) \in \mathrm{dom}(B_h^k) \subset B_h^{k,\mathtt{pa}}$, we have by Lemma 5

$$q_1 \leq \alpha^0_{n_h^k(B_h^{k,\mathtt{pa}})} H + \beta_{n_h^k(B_h^{k,\mathtt{pa}})} + 4L \cdot \mathtt{rad}(B_h^{k,\mathtt{pa}}) + \sum_{i=1}^{n_h^k(B_h^{k,\mathtt{par}})} \alpha^i_{n_h^k(B_h^{k,\mathtt{pa}})} (V_{h+1}^{k_i(B_h^{k,\mathtt{pa}})} - V^*_{h+1})(x_{h+1}^{k_i(B_h^{k,\mathtt{pa}})})$$

where we denote by $k_i(B) \in [1, n_h^k(B)]$ the i-th episode where B was selected by the algorithm at step h. As $B_h^{k,\mathtt{pa}}$ is a parent, we have $n_h^k(B_h^{k,\mathtt{pa}}) > 0$ implying that $\alpha^0_{n_h^k(B_h^{k,\mathtt{pa}})} = \mathbb{I}\{n_h^k(B_h^{k,\mathtt{pa}}) = 0\} = 0$. Moreover, by the activation rule, we have $\frac{1}{\sqrt{n_h^k(B_h^{k,\mathtt{pa}})}} \leq \mathtt{rad}(B_h^{k,\mathtt{pa}})$, implying that $\beta_{n_h^k(B_h^{k,\mathtt{pa}})} \leq 16\sqrt{\frac{H^3\iota}{n_h^k(B_h^{k,\mathtt{pa}})}} \leq 16\sqrt{H^3\iota}\,\mathtt{rad}(B_h^{k,\mathtt{pa}}) = 32\sqrt{H^3\iota}\,\mathtt{rad}(B_h^k)$. Consequently,

$$q_1 \leq (8L + 32\sqrt{H^3\iota})\,\mathtt{rad}(B_h^k) + \sum_{i=1}^{n_h^k(B_h^{k,\mathtt{pa}})} \alpha^i_{n_h^k(B_h^{k,\mathtt{pa}})} \phi_{h+1}^{k_i(B_h^{k,\mathtt{pa}})}$$

and therefore,

$$\delta_h^k \leq (11L + 32\sqrt{H^3\iota})\,\mathtt{rad}(B_h^k) + \sum_{i=1}^{n_h^k(B_h^{k,\mathtt{pa}})} \alpha^i_{n_h^k(B_h^{k,\mathtt{pa}})} \phi_{h+1}^{k_i(B_h^{k,\mathtt{pa}})} + (Q_h^* - Q_h^{\pi_k})(s_h^k, a_h^k) \tag{C.1.9}$$

2. **Case of $B_h^k = B_{\mathtt{init}}$:**

$$\begin{aligned}
\delta_h^k &\leq \max_{B \in \mathtt{rel}_h^k(s_h^k)} \mathtt{index}_h^k(B) - V_h^{\pi_k}(s_h^k) \\
&= \mathtt{index}_h^k(B_h^k) - Q_h^{\pi_k}(s_h^k, a_h^k) \\
&\leq L \cdot \mathtt{rad}(B_h^k) + \widehat{\mathrm{Q}}_h^k(B_h^k) - Q_h^{\pi_k}(s_h^k, a_h^k) \\
&= L \cdot \mathtt{rad}(B_h^k) + \widehat{\mathrm{Q}}_h^k(B_h^k) - Q_h^*(s_h^k, a_h^k) + (Q_h^* - Q_h^{\pi_k})(s_h^k, a_h^k) \\
&\leq 5L \cdot \mathtt{rad}(B_h^k) + \alpha^0_{n_h^k(B_h^k)} H + \beta_{n_h^k(B_h^k)} + \sum_{i=1}^{n_h^k(B_h^k)} \alpha^i_{n_h^k(B_h^k)} \phi_{h+1}^{k_i(B_h^k)} + (Q_h^* - Q_h^{\pi_k})(s_h^k, a_h^k) \\
&\leq \alpha^0_{n_h^k(B_h^k)} H + (5L + 16\sqrt{H^3\iota})\,\mathtt{rad}(B_h^k) + \sum_{i=1}^{n_h^k(B_h^k)} \alpha^i_{n_h^k(B_h^k)} \phi_{h+1}^{k_i(B_h^k)} + (Q_h^* - Q_h^{\pi_k})(s_h^k, a_h^k)
\end{aligned} \tag{C.1.10}$$

The third inequality follows from lemma 5 and the last inequality follows from the fact that $\texttt{rad}(B_h^k) = \texttt{rad}(B_{\texttt{init}}) = 1$

Now, we can unify the bound (C.1.9) obtained in the first case where the algorithm selects a ball other than the initial ball and the bound (C.1.10) in second case where the initial ball is selected. To do that, we consider, by abuse of notation, that the initial ball is parent of itself i.e when $B_h^k = B_{\texttt{init}}$ we have $B_h^{k,\texttt{pa}} = B_{\texttt{init}}$ and we take the maximum over the two bounds

$$\begin{aligned}\delta_h^k &\leq \alpha^0_{n_h^k(B_h^k)} H \mathbb{I}_{\{B_h^k = B_{\texttt{init}}\}} + (11L + 32\sqrt{H^3\iota})\,\texttt{rad}(B_h^k) \\ &\quad + \sum_{i=1}^{n_h^k(B_h^{k,\texttt{pa}})} \alpha^i_{n_h^k(B_h^{k,\texttt{pa}})} \phi_{h+1}^{k_i(B_h^{k,\texttt{pa}})} + (Q_h^* - Q_h^{\pi_k})(s_h^k, a_h^k) \qquad \text{(C.1.11)}\end{aligned}$$

we obtain the desired result but noting that

$$\begin{aligned}&(Q_h^\star - Q_h^{\pi_k})(s_h^k, a_h^k) \\ &= [\mathbb{P}_h(V_h^\star - V_h^{\pi_k})](s_h^k, a_h^k) \\ &= [(P_h - \widehat{P}_h)(V_{h+1}^\star - V_{h+1}^{\pi_k})](s_h^k, a_h^k) + (V_{h+1}^\star - V_{h+1}^{\pi_k})(s_{h+1}^k) \\ &= [(P_h - \widehat{P}_h)(V_{h+1}^\star - V_{h+1}^{\pi_k})](s_h^k, a_h^k) + (V^\star - \widehat{\mathrm{V}}^{\pi_k})(s_{h+1}^k) + (\widehat{\mathrm{V}}_{h+1}^k - V_{h+1}^{\pi_k})(s_{h+1}^k) \\ &= \xi_{h+1}^k - \phi_{h+1}^k + \delta_{h+1}^k\end{aligned}$$

C.1.5 Bounding $\sum_{h=1}^H \sum_{k=1}^K \xi_{h+1}^k$

Lemma 12. *With probability at least $1 - p/2$, we have*

$$\sum_{h=1}^H \sum_{k=1}^K \xi_{h+1}^k \leq 4\sqrt{2H^3K\iota}$$

Let $\mathcal{F}_{k,h}$ be the σ-field generated by all the random variables until episode k, step h. Then, $\xi_{h+1}^k = [(P_h - \widehat{P}_h)(V_{h+1}^* - V_{h+1}^{\pi_k})](s_h^k, a_h^k)$ is a martingale difference sequence w.r.t the filtration $\{\mathcal{F}_{k,h}\}_{k,h\geq 0}$ bounded by $4H$. By Azuma-Hoeffding (lemma 15), we have $\forall t > 0, \Pr\left[\left|\sum_{h=1}^H \sum_{k=1}^K \xi_{h+1}^k\right| \geq t\right] \leq 2\exp\left(\frac{-t^2}{32H^3K}\right)$ Therefore,

$$\begin{aligned}\Pr\left[\left|\sum_{h=1}^H \sum_{k=1}^K \xi_{h+1}^k\right| \geq 4\sqrt{2H^3K\iota}\right] &\leq 2\exp\left(\frac{-32H^3K\iota}{32H^3K}\right) \\ &= 2\frac{p}{4H^2K^2} \leq p/2\end{aligned}$$

Hence, with probability at least $1 - p/2$, we have

$$\sum_{h=1}^{H}\sum_{k=1}^{K}\xi_{h+1}^{k} \le 4\sqrt{2H^3K\iota}$$

C.2 Misspecified Setting: Approximately Lipschtiz Case

The proof structure is similar to the structure in Appendix B. We will particularly focus on the parts that require different treatments in the misspecified setting.

C.2.1 Recursive Formula of $\widehat{Q}_h^k(B) - Q_h^\star(s,a)$

Let $B \in \mathcal{B}_h^k$ and $(s,a) \in \texttt{dom}_h^k(B)$.

Since $\sum_{i=0}^t \alpha_t^i = 1$, we have that $Q_h^\star(s,a) = \alpha_t^0 Q_h^\star(s,a) + \sum_{i=1}^t \alpha_t^i Q_h^*(s,a)$.

By the ϵ-approximately Lipschitz assumption 5 and the fact $\forall i \in [t], (s_h^{k_i}, a_h^{k_i}) \in B$ and $(s,a) \in B$, we have:

$$|Q_h^\star(s_h^{k_i}, a_h^{k_i}) - Q_h^\star(s,a)| \le L \cdot \texttt{dist}((x_h^{k_i}, a_h^{k_i}), (x,a)) + 2\epsilon \le 2L \cdot \texttt{rad}(B) + 2\epsilon.$$

Then we have

$$Q_h^\star(s,a) \ge \alpha_t^0 Q_h^\star(s,a) + \sum_{i=1}^t \alpha_t^i \left(Q_h^\star(s_h^{k_i}, a_h^{k_i}) - 2L \cdot \texttt{rad}(B) - 2\epsilon\right) \tag{C.2.1}$$

$$Q_h^\star(s,a) \le \alpha_t^0 Q_h^\star(s,a) + \sum_{i=1}^t \alpha_t^i \left(Q_h^\star(s_h^{k_i}, a_h^{k_i}) + 2L \cdot \texttt{rad}(B) + 2\epsilon\right). \tag{C.2.2}$$

By Bellman equation, we have $Q_h^\star(s_h^{k_i}, a_h^{k_i}) = r_h(s_h^{k_i}, a_h^{k_i}) + [P_h V_{h+1}^\star](s_h^{k_i}, a_h^{k_i})$. Recall $[\widehat{P}_h^{k_i} V_{h+1}](s_h^{k_i}, a_h^{k_i}) = V_{h+1}(s_{h+1}^{k_i})$, we have:

$$Q_h^\star(s_h^{k_i}, a_h^{k_i}) = r_h(s_h^{k_i}, a_h^{k_i}) + [(P_h - \widehat{P}_h^{k_i})V_{h+1}^\star](s_h^{k_i}, a_h^{k_i}) + V_{h+1}^\star(s_{h+1}^{k_i}).$$

Substitute the above equality into Eq. (C.2.1) and (C.2.2), we have:

$$\begin{aligned} Q_h^\star(s,a) \ge \alpha_t^0 Q_h^\star(s,a)) + \sum_{i=1}^t \alpha_t^i \Big(& r_h(x_h^{k_i}, a_h^{k_i}) + [(P_h - \widehat{P}_h^{k_i})V_{h+1}^\star](x_h^{k_i}, a_h^{k_i}) \\ & + V_{h+1}^\star(x_{h+1}^{k_i}) - 2L \cdot \texttt{rad}(B) - 2\epsilon \Big) \end{aligned}$$

$$Q_h^\star(s,a) \le \alpha_t^0 Q_h^\star(s,a)) + \sum_{i=1}^t \alpha_t^i \Big(r_h(x_h^{k_i}, a_h^{k_i}) + [(P_h - \widehat{P}_h^{k_i}) V_{h+1}^\star](x_h^{k_i}, a_h^{k_i}) + V_{h+1}^\star(x_{h+1}^{k_i}) + 2L \cdot \mathtt{rad}(B) + 2\epsilon \Big)$$

Subtracting the formula in Lemma 4 from the two above inequalities, we have:

$$\widehat{\mathrm{Q}}_h^k(B) - Q_h^\star(s,a) \ge \sum_{i=1}^t \alpha_t^i \Big((\widehat{\mathrm{V}}_{h+1}^{k_i} - V_{h+1}^\star)(x_{h+1}^{k_i}) + [(\widehat{P}_h^{k_i} - P_h) V_{h+1}^*](x_h^{k_i}, a_h^{k_i}) + u_i - 2\epsilon \Big) \tag{C.2.3}$$

$$\widehat{\mathrm{Q}}_h^k(B) - Q_h^\star(s,a) \le \alpha_t^0 H + \sum_{i=1}^t \alpha_t^i \Big((\widehat{\mathrm{V}}_{h+1}^{k_i} - V_{h+1}^\star)(x_{h+1}^{k_i}) + [(\widehat{P}_h^{k_i} - P_h) V_{h+1}^*](x_h^{k_i}, a_h^{k_i}) + u_i + 4L \cdot \mathtt{rad}(B) + 2\epsilon \Big) \tag{C.2.4}$$

C.2.2 Bounding of $Q_h^k(B) - Q_h^*(s,a)$

Lemma 13. *Suppose Assumption 5 holds. For any $p \in (0,1)$, we have $\beta_t = 2\sum_{i=1}^t \alpha_t^i u_i \le 16\sqrt{\frac{H^3 \iota}{t}}$ and, with probability at least $1 - p/2$, we have that for all $(s,a,h,k) \in \mathcal{S} \times \mathcal{A} \times [H] \times [K]$ and any ball B such that $(s,a) \in dom_h^k(B)$:*

(a) $\widehat{\mathrm{Q}}_h^k(B) - Q_h^\star(s,a) \ge -4(H-h+1)\epsilon$

(b) $\widehat{\mathrm{Q}}_h^k(B) - Q_h^\star(x,a) \le \alpha_t^0 \cdot H + \beta_t + 4L \cdot \mathtt{rad}(B) + 2\epsilon + \sum_{i=1}^t \alpha_t^i \cdot (\widehat{\mathrm{V}}_{h+1}^{k_i} - V_{h+1}^*)(s_{h+1}^{k_i})$

where $t = n_h^k(B)$ and $k_1, \cdots, k_t < k$ are the episodes where B was selected at step h.

C.2.3 High Probability Bound On The Sampling Noise

The same reasoning as in the subsection C.1.3 in the exact lipschitz case gives: with probability at least $1 - p/2$: $\forall (h,k) \in [H] \times [K]$ and for all ball $B \in \mathcal{B}_h^k$,

$$\left| \sum_{i=1}^t \alpha_t^i \cdot [(\widehat{P}_h^{k_i} - P_h) V_{h+1}^*](x_h^{k_i}, a_h^{k_i}) \right| \le \beta_t/2, \text{where } t = n_h^k(B) \tag{C.2.5}$$

C.2.4 Approximate Optimism Of Q-values

We proceed by induction. By definition, we have $\widehat{\mathrm{Q}}^k_{H+1} = Q^*_{H+1} = 0$ which implies $Q^k_{H+1}(B) - Q^*_{H+1}(s,a) = -4(H-(H+1)+1)\epsilon$. Assume that $Q^k_{h+1}(B) - Q^*_{h+1}(s,a) \geq -4(H-(h+1)+1)\epsilon = -4(H-h)\epsilon$.

Let $i \in [1,t]$, We have $V^\star_{h+1}(s^{k_i}_{h+1}) = Q^\star_{h+1}(s^{k_i}_{h+1}, \pi^\star_{h+1}(s^{k_i}_{h+1}))$. As the set of domains of active balls covers the entire space, there exists $B^\star \in \mathcal{B}^k_{h+1}$ such that $(s^{k_i}_{h+1}, \pi^\star_{h+1}(s^{k_i}_{h+1})) \in \mathtt{dom}^k_{h+1}(B^*)$. By the definition of index, we have $\mathtt{index}^k_{h+1}(B^\star) = L \cdot \mathtt{rad}(B^\star) + \widehat{\mathrm{Q}}^k_{h+1}(\tilde{B}^\star) + L \cdot \mathtt{dist}(\tilde{B}^\star, B^\star)$ for some ball $\tilde{B}^\star$.

We have

$$
\begin{aligned}
&\widehat{\mathrm{V}}^k_{h+1}(s^{k_i}_{h+1}) - V^\star_{h+1}(s^{k_i}_{h+1}) \\
&= \min\{H, \max_{B \in \mathtt{rel}^k_{h+1}(s^{k_i}_{h+1})} \mathtt{index}^k_{h+1}(B)\} - Q^\star_{h+1}(s^{k_i}_{h+1}, \pi^\star(s^{k_i}_{h+1})) \\
&\geq \max_{B \in \mathtt{rel}^k_{h+1}(s^{k_i}_{h+1})} \mathtt{index}^k_{h+1}(B) - Q^\star_{h+1}(s^{k_i}_{h+1}, \pi^\star(s^{k_i}_{h+1})) \\
&\geq \mathtt{index}^k_{h+1}(B^\star) - Q^\star_{h+1}(s^{k_i}_{h+1}, \pi^\star(s^{k_i}_{h+1})) \\
&= L \cdot \mathtt{rad}(B^\star) + \widehat{\mathrm{Q}}^k_{h+1}(\tilde{B}^\star) + L \cdot \mathtt{dist}(\tilde{B}^\star, B^\star) - Q^\star_{h+1}(s^{k_i}_{h+1}, \pi^\star(s^{k_i}_{h+1})) \\
&\geq L \cdot \mathtt{rad}(B^\star) - 4(H-h)\epsilon + Q^\star_{h+1}(s_{\tilde{B}^\star}, a_{\tilde{B}^\star}) + L \cdot \mathtt{dist}(\tilde{B}^\star, B^\star) - Q^\star_{h+1}(s^{k_i}_{h+1}, \pi^\star(s^{k_i}_{h+1})) \\
&\geq L \cdot \mathtt{rad}(B^\star) - 4(H-h)\epsilon + Q^\star_{h+1}(s_{B^\star}, a_{B^\star}) - 2\epsilon - Q^\star_{h+1}(s^{k_i}_{h+1}, \pi^\star(s^{k_i}_{h+1})) \\
&\geq -(4(H-h)+2)\epsilon
\end{aligned}
$$

Where $(s_{B^\star}, a_{B^\star})$ and $(s_{\tilde{B}^\star}, a_{\tilde{B}^\star})$ denote respectively the centers of balls $B^\star$ and $\tilde{B}^\star$. The first inequality follows from $Q^\star_{h+1}(s,a) \leq H$ for any state-action pair (s,a). The third inequality follows from the induction hypothesis. The fourth and the last inequalities follow from the assumption 5

Therefore, we have

$$
\begin{aligned}
Q^k_h(B) - Q^*_h(s,a) &\geq \sum_{i=1}^t \alpha^i_t \cdot \Big((\widehat{\mathrm{V}}^{k_i}_{h+1} - V^\star_{h+1})(s^{k_i}_{h+1}) + [(\widehat{P}^{k_i}_h - P_h)V^\star_{h+1}](s^{k_i}_h, a^{k_i}_h) + u_i - 2\epsilon \Big) \\
&\geq -(4(H-h)+2)\epsilon - \beta_t/2 + \beta_t/2 - 2\epsilon = -4(H-h+1)\epsilon
\end{aligned}
$$

C.2.5 Upper Bound of $\widehat{\mathrm{Q}}_h^k(B) - Q_h^\star(s,a)$

We have:

$$
\begin{aligned}
&\widehat{\mathrm{Q}}_h^k(B) - Q_h^\star(s,a) \\
&\leq \alpha_t^0 \cdot H + \sum_{i=1}^{t} \alpha_t^i \cdot \Big((V_{h+1}^{k_i} - V_{h+1}^*)(x_{h+1}^{k_i}) + [(\widehat{P}_h^{k_i} - P_h)V_{h+1}^*](x_h^{k_i}, a_h^{k_i}) + u_i + 4\,\mathtt{rad}(B) + 2\epsilon \Big) \\
&\leq \alpha_t^0 H + \sum_{i=1}^{t} \alpha_t^i \cdot (\widehat{\mathrm{V}}_{h+1}^{k_i} - V_{h+1}^*)(x_{h+1}^{k_i}) + \beta_t/2 + \sum_{i=1}^{t} \alpha_t^i u_i + \sum_{i=1}^{t} \alpha_t^i (4L \cdot \mathtt{rad}(B) + 2\epsilon) \\
&\leq \alpha_t^0 H + \beta_t + 4L \cdot \mathtt{rad}(B) + 2\epsilon + \sum_{i=1}^{t} \alpha_t^i \cdot (\widehat{\mathrm{V}}_{h+1}^{k_i} - V_{h+1}^*)(x_{h+1}^{k_i})
\end{aligned}
$$

where the second inequality follows from the inequality C.2.5. The third inequality follows from $\sum_{i=1}^{t} \alpha_t^i \leq 1$

Lemma 14 (Approximate Optimism)**.** *Following the same setting as in Lemma 13, for any (h,k), with probability at least $1 - p/2$, we have for any $s \in \mathcal{S}$:*

$$\widehat{\mathrm{V}}_h^k(s) \geq V_h^\star(s) - (4(H-h)+5)\epsilon$$

Proof. Let $s \in \mathcal{S}$, We have $V_h^\star(s) = Q_h^\star(s, \pi_h^\star(s))$. As the set of domains of active balls covers the entire space, there exists $B^\star \in \mathcal{B}_{h+1}^k$ such that $(s, \pi_h^\star(s)) \in \mathtt{dom}_h^k(B^*)$. By the definition of index, we have $\mathtt{index}_h^k(B^\star) = L \cdot \mathtt{rad}(B^\star) + \widehat{\mathrm{Q}}_h^k(\tilde{B}^\star) + L \cdot \mathtt{dist}(\tilde{B}^\star, B^\star)$ for some active ball $\tilde{B}^\star$.

We have

$$
\begin{aligned}
&\widehat{\mathrm{V}}_h^k(s) - V_h^\star(s) \\
&= \min\{H, \max_{B \in \mathtt{rel}_h^k(s)} \mathtt{index}_h^k(B)\} - Q_h^\star(s, \pi^\star(s)) \\
&\geq \max_{B \in \mathtt{rel}_h^k(s)} \mathtt{index}_h^k(B) - Q_h^\star(s, \pi^\star(s)) \\
&\geq \mathtt{index}_h^k(B^\star) - Q_h^\star(s, \pi^\star(s)) \\
&= L \cdot \mathtt{rad}(B^\star) + \widehat{\mathrm{Q}}_h^k(\tilde{B}^\star) + L \cdot \mathtt{dist}(\tilde{B}^\star, B^\star) - Q_h^\star(s, \pi^\star(s)) \\
&\geq L \cdot \mathtt{rad}(B^\star) + Q_h^\star(s_{\tilde{B}^\star}, a_{\tilde{B}^\star}) - 4(H-h+1)\epsilon + L \cdot \mathtt{dist}(\tilde{B}^\star, B^\star) - Q_h^\star(s, \pi^\star(s)) \\
&\geq L \cdot \mathtt{rad}(B^\star) + Q_h^\star(s_{B^\star}, a_{B^\star}) - 2\epsilon - 4(H-h+1)\epsilon - Q_h^\star(s, \pi^\star(s)) \\
&\geq -4\epsilon - 4(H-h+1)\epsilon = -(4(H-h)+5)\epsilon
\end{aligned}
$$

Where $(s_{B^\star}, a_{B^\star})$ and $(s_{\tilde{B}^\star}, a_{\tilde{B}^\star})$ denote respectively the centers of balls $B^\star$ and $\tilde{B}^\star$. The first inequality follows from $Q_h^\star(s,a) \leq H$ for any state-action pair (s,a). The third inequality follows from lemma 13. The fourth and the last inequalities follow from assumption 5 □

C.2.6 Regret Analysis

By the approximate optimism of our estimates with respect to the true value function (see lemma 14), we have with probability at least $1 - p/2$

$$\begin{aligned}\textsc{Regret}(K) &= \sum_{k=1}^{K} (V_1^* - V_1^{\pi_k})(x_1^k) \\ &\leq \sum_{k=1}^{K} (\widehat{\mathrm{V}}_1^k - V_1^{\pi_k})(x_1^k) + K(4H+1)\epsilon \\ &= \sum_{k=1}^{K} \delta_1^k + K(4H+1)\epsilon\end{aligned}$$

Similarly to the Lemma 7, we can show that using Lemma 13 applied on $B_h^{k,\mathtt{pa}}$ the parent of the selected ball at step h of the episode k.

$$\begin{aligned}\delta_h^k \leq \alpha^0_{n_h^k(B_h^k)} H \mathbb{I}_{\{B_h^k = B_{\mathtt{init}}\}} + (11L + 32\sqrt{H^3\imath})\,\mathtt{rad}(B_h^k) \\ + \sum_{i=1}^{n_h^k(B_h^{k,\mathtt{pa}})} \alpha^i_{n_h^k(B_h^{k,\mathtt{pa}})} \phi_{h+1}^{k_i(B_h^{k,\mathtt{pa}})} - \phi_{h+1}^k + \delta_{h+1}^k + \xi_{h+1}^k + 2\epsilon\end{aligned}$$

Following the same steps of the Section 5.7.1 in the exact lipschitz setting, we obtain

$$\sum_{k=1}^{K} \delta_h^k \leq H + (11L + 32\sqrt{H^3\imath}) \sum_{k=1}^{K} \mathtt{rad}(B_h^k) + \left(1 + \frac{1}{H}\right) \sum_{k=1}^{K} \delta_{h+1}^k + \sum_{k=1}^{K} \xi_{h+1}^k + 2K\epsilon$$

By unrolling the last inequality for $h \in [H]$ and using the fact $\delta_{H+1}^k = 0 \quad \forall k \in [K]$, we obtain

$$\begin{aligned}\sum_{k=1}^{K} \delta_1^k &\leq \sum_{h=1}^{H} (1 + \frac{1}{H})^{h-1} \left(H + (11L + 32\sqrt{H^3\imath}) \sum_{k=1}^{K} \mathtt{rad}(B_h^k) + \sum_{k=1}^{K} \xi_{h+1}^k + 2K\epsilon \right) \\ &\leq 3H^2 + 3(11L + 32\sqrt{H^3\imath}) \sum_{h=1}^{H} \sum_{k=1}^{K} \mathtt{rad}(B_h^k) + 3 \sum_{h=1}^{H} \sum_{k=1}^{K} \xi_{h+1}^k + 6HK\epsilon\end{aligned} \tag{C.2.6}$$

Plugging bounds (5.7.2) and (5.7.3) from Section 5.7.1 in (C.2.6) and using union bound, we have with probability $1 - p$

$$\sum_{k=1}^{K} \delta_1^k \leq O\left(H^2 + \sqrt{H^3 K\imath} + (L + \sqrt{H^5\imath}) \min_{r_0 \in (0,1)} \left\{ K r_0 + \sum_{\substack{r = 2^{-i} \\ r \geq r_0}} \frac{M(r)}{r} \right\} + HK\epsilon \right)$$

We obtain the regret bound in theorem 3 by noting that $\textsc{Regret}(K) \leq \sum_{k=1}^{K} \delta_1^k + O(HK\epsilon)$.

C.3 Technical Lemmas

Lemma 15 (Azuma-Hoeffding inequality)**.** *Suppose* $\{X_k : k = 0, 1, 2, 3, \cdots\}$ *is a martingale and* $|X_k - X_{k-1}| < c_k$*, almost surely. Then for all positive integers* N *and all positive reals* t*,*

$$\Pr[|X_N - X_0| \geq t] \leq 2 \exp\left(\frac{-t^2}{2\sum_{k=1}^{N} c_k^2}\right).$$

Lemma 16 (Lemma 4.1 in Jin et al. [2018])**.** *The following properties hold for* α_t^i*:*

(a) $\frac{1}{\sqrt{t}} \leq \sum_{i=1}^{t} \frac{\alpha_t^i}{\sqrt{i}} \leq \frac{2}{\sqrt{t}}$ *for every* $t \geq 1$*.*

(b) $\max_{i \in [t]} \alpha_t^i \leq \frac{2H}{t}$ *and* $\sum_{i=1}^{t} (\alpha_t^i)^2 \leq \frac{2H}{t}$ *for every* $t \geq 1$*.*

(c) $\sum_{t=i}^{\infty} \alpha_t^i = 1 + \frac{1}{H}$ *for every* $i \geq 1$*.*

C.3.1 Few Reminders on Probability Theory

We consider a probability space $(\Omega, \mathcal{F}, P)$. We borrow notation from Qian et al. [2018]. We call filtration any increasing (for the inclusion) sequence of sub-σ-algebras of $\mathcal{F}$ i.e., $(\mathcal{F}_n)_{n \in \mathbb{N}}$ where $\forall n \in \mathbb{N}$, $\mathcal{F}_n \subset \mathcal{F}_{n+1} \subset \mathcal{F}$. We denote by $\mathcal{F}_\infty = \cup_{n \in \mathbb{N}} \mathcal{F}_n$.

Definition 7 (Stopping time)**.** *A random variable* $\tau : \Omega :\to \mathbb{N} \cup \{+\infty\}$ *is called stopping time with respect to a filtration* $(\mathcal{F}_n)_{n \in \mathbb{N}}$ *if for all* $n \in \mathbb{N}$*,* $\{\tau = n\} \in \mathcal{F}_n$*.*

Definition 8 (σ-algebra at stopping time)**.** *let* τ *be a stopping time. An event prior to* τ *is any event* $A \in \mathcal{F}_\infty$ *s.t* $A \cap \tau = n \in \mathcal{F}_n$ *for all* $n \in \mathbb{N}$*. The set of events prior to* τ *is a* σ*-algebra denoted* $\mathcal{F}_\tau$ *and called* $\sigma-$*algebra at time* τ*:*

$$\mathcal{F}_\tau = \{A \in \mathcal{F}_\infty, \forall \in \mathbb{N}, A \cap \tau = n \in \mathcal{F}_n\}$$

Lemma 17. *let* τ_1 *and* τ_2 *be two stopping times with respect to the same filtration* $(\mathcal{F}_n)_{n \in \mathbb{N}}$ *s.t* $\tau_1 \leq \tau_2$ *almost surely. Then* $\mathcal{F}_{\tau_1} \subset \mathcal{F}_{\tau_2}$

D Appendix for Chapter 6

D.1 Technical Gaps in Published Bandit Papers

In this section, we highlight the technical error made by [Cheung et al., 2019] when controlling the bias term due to the non-stationarity of the reward function. Let us first recall the non-stationary linear bandit model

Definition 9 (Non-stationary linear bandit)**.** *At iteration t, the player makes a decision A_t from a feasible set $\mathcal{A} \subset \mathbb{R}^d$, then observes the reward r_t satisfying:*

$$r_t = A_t^\top \boldsymbol{\theta}_t + z_t \tag{D.1.1}$$

where $\boldsymbol{\theta}_t$ is the unknown regression parameter at iteration t and z_t is conditionally σ-subgaussian noise. We assume further that $\|A\| \leq 1, \forall A \in \mathcal{A}$ and $\|\boldsymbol{\theta}_t\| \leq S, \forall t$.

Cheung et al. [2019] propose the SW-UCB algorithm based on a sliding window approach of size W. At time t, actions are selected as follows:

$$A_t = \arg\max_{a \in \mathcal{A}} a^\top \hat{\boldsymbol{\theta}}_t + \beta \|a\|_{V_t^{-1}} \tag{D.1.2}$$

where $\hat{\boldsymbol{\theta}}_t$ is the solution of the sliding window least squares problem:

$$\hat{\boldsymbol{\theta}}_t = V_t^{-1} \sum_{\tau=\max\{1,t-W\}}^{t-1} A_\tau r_\tau \quad \text{, where} \quad V_t = \sum_{\tau=\max\{1,t-W\}}^{t-1} A_\tau A_\tau^\top + \lambda \cdot \mathbf{I} \text{ is the Gram matrix.} \tag{D.1.3}$$

In the proof of lemma 1 in Cheung et al. [2019], the authors consider matrix $M = V_t^{-1} X$ where $X = \sum_{\tau=t-W}^{p} A_\tau A_\tau^\top$ for any integer $p \in \{t-W, \dots, t-1\}$. They attempt to show that M is positive semi-definite (PSD) (i.e $y^\top M y \geq 0, \forall y \in \mathbb{R}^d$) as follows: they first prove that M shares the same characteristic polynomial as the matrix $V_t^{-1/2} X V^{-1/2}$, then assert that since $V_t^{-1/2} X V^{-1/2}$ is PSD, M is PSD as well.

Unfortunately, this last assertion does not hold in general. As a counterexample, let us consider the 2 dimensional identity matrix $\mathbf{I}$ and $B = ((1,0)^\top, (-10,1)^\top)$. $\mathbf{I}$ and B share the same characteristic polynomial

$p(x) = (x-1)^2$, $\mathbf{I}$ is obviously PSD but B is not, as for $y = (1,1)^\top$, we have $y^\top By = -8 < 0$.

Moreover, in general, a matrix of the form $M = V_t^{-1}X$ is not guaranteed to be PSD. If one sets $d = 2, t = 3, \lambda = 1, A_1 = (1,0)^\top$ and $A_2 = (1,1)^\top$. with $A_1 = (1,0)^\top$ and $A_2 = (1,1)^\top$, we have $M = V_t^{-1}A_1A_1^\top = ((0.4,-0.2)^\top, (0,0)^\top)$. If we consider $y = (1,5)^\top$, we have $y^\top My = -0.6 < 0$.

D.2 Regret Reanalysis of D-LinUCB

Russac et al. [2019] propose the D-LinUCB algorithm, based on sequential weighted least squares regression. At time t, actions are selected as follows:

$$x_t = \arg\max_{x \in \mathcal{X}_t} x^\top \hat{\boldsymbol{\theta}}_t + \beta \|x\|_{V_t^{-1}\tilde{V}_t V_t^{-1}} \tag{D.2.1}$$

where $V_t = \sum_{\tau=1}^{t-1} \eta^{-\tau} x_\tau x_\tau^\top + \lambda\eta^{-(t-1)} \cdot \mathbf{I}$ is the Gram matrix, $\tilde{V}_t = \sum_{\tau=1}^{t-1} \eta^{-2\tau} x_\tau x_\tau^\top + \lambda\eta^{-2(t-1)} \cdot \mathbf{I}$ and $\hat{\boldsymbol{\theta}}_t$ is the solution the weighted least squares problem:

$$\hat{\boldsymbol{\theta}}_t = V_t^{-1} \sum_{\tau=1}^{t-1} \eta^{-\tau} x_\tau r_\tau. \tag{D.2.2}$$

As our analysis follows the same proof steps as in Russac et al. [2019], we will only highlight our proposed fix to their technical error and the changes that it induces.

Non-stationarity bias Let $\bar{\boldsymbol{\theta}}_t \triangleq V_t^{-1} \sum_{\tau=1}^{t-1} \eta^{-\tau} x_\tau x_\tau^\top \boldsymbol{\theta}_\tau + \lambda\eta^{-(t-1)}\boldsymbol{\theta}_t$ the weighted average of the true regression parameters. To characterize the bias, Russac et al. [2019] attempt to control directly $\|\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}}_t\|$. Instead, we propose to control $|x^\top(\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}}_t)|$ for any $x \in \mathcal{X}$ and then use the fact that $\|\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}}_t\| = \max_{x:\|x\|=1} |x^\top(\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}}_t)|$

$$\begin{aligned} |x^\top(\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}}_t)| &= \left| x^\top V_t^{-1} \sum_{\tau=1}^{t-1} \eta^{-\tau} x_\tau x_\tau^\top (\boldsymbol{\theta}_\tau - \boldsymbol{\theta}_t) \right| \\ &\leq \underbrace{\left| x^\top V_t^{-1} \sum_{\tau=t-W}^{t-1} \eta^{-\tau} x_\tau x_\tau^\top (\boldsymbol{\theta}_\tau - \boldsymbol{\theta}_t) \right|}_{(\star)} + \underbrace{\left| x^\top V_t^{-1} \sum_{\tau=1}^{t-W-1} \eta^{-\tau} x_\tau x_\tau^\top (\boldsymbol{\theta}_\tau - \boldsymbol{\theta}_t) \right|}_{(\star\star)} \end{aligned}$$

Bound on $(\star)$:

$$
\begin{aligned}
&\left| x^\top V_t^{-1} \sum_{\tau=t-W}^{t-1} \eta^{-\tau} x_\tau x_\tau^\top (\boldsymbol{\theta}_\tau - \boldsymbol{\theta}_t) \right| \\
&\leq \sum_{\tau=t-W}^{t-1} \eta^{-\tau} \left| x^\top V_t^{-1} x_\tau \right| \cdot |x_\tau^\top (\boldsymbol{\theta}_\tau - \boldsymbol{\theta}_t)| && \text{(triangle inequality)} \\
&= \sum_{\tau=t-W}^{t-1} \eta^{-\tau} |x^\top V_t^{-1} x_\tau| \cdot |x_\tau^\top (\sum_{s=\tau}^{t-1} (\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}))| \\
&\leq \sum_{\tau=t-W}^{t-1} \eta^{-\tau} |x^\top V_t^{-1} x_\tau| \cdot \|x_\tau\| \cdot \| \sum_{s=\tau}^{t-1} (\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}) \| && \text{(Cauchy-Schwarz)} \\
&\leq \sum_{\tau=t-W}^{t-1} \eta^{-\tau} |x^\top V_t^{-1} x_\tau| \cdot \sum_{s=\tau}^{t-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\| && (\|x_\tau\| \leq 1) \\
&\leq \sum_{s=t-W}^{t-1} \sum_{\tau=t-W}^{s} \eta^{-\tau} |x^\top V_t^{-1} x_\tau| \cdot \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\| \\
& && (\textstyle\sum_{\tau=t-W}^{t-1} \sum_{s=\tau}^{t-1} = \sum_{s=t-W}^{t-1} \sum_{\tau=t-W}^{s}) \\
&\leq \sum_{s=t-W}^{t-1} \sqrt{\left[\sum_{\tau=t-W}^{s} \eta^{-\tau} x^\top V_t^{-1} x \right] \cdot \left[\sum_{\tau=t-W}^{s} \eta^{-\tau} x_\tau^\top V_t^{-1} x_\tau \right]} \cdot \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\| \\
& && \text{(Cauchy-Schwarz)} \\
&\leq \sum_{s=t-W}^{t-1} \sqrt{\left[\sum_{\tau=t-W}^{s} \eta^{-\tau} x^\top V_t^{-1} x \right] \cdot d} \cdot \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\| && \text{(by lemma 22)} \\
&\leq \|x\| \sqrt{d} \sum_{s=t-W}^{t-1} \sqrt{\frac{\sum_{\tau=t-W}^{t-1} \eta^{-\tau}}{\lambda \eta^{-(t-1)}}} \cdot \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\| && (\lambda_{\max}(V_t^{-1}) \leq \tfrac{1}{\lambda \eta^{-(t-1)}}) \\
&\leq \|x\| \sqrt{\frac{d}{\lambda(1-\eta)}} \sum_{s=t-W}^{t-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|
\end{aligned}
$$

Bound on $(\star\star)$:

$$\begin{aligned}
\left| x^\top V_t^{-1} \sum_{\tau=1}^{t-W-1} \eta^{-\tau} x_\tau x_\tau^\top (\boldsymbol{\theta}_\tau - \boldsymbol{\theta}_t) \right| &\leq \|x\| \left\| V_t^{-1} \sum_{\tau=1}^{t-W-1} \eta^{-\tau} x_\tau x_\tau^\top (\boldsymbol{\theta}_\tau - \boldsymbol{\theta}_t) \right\| \\
&\leq \|x\| \frac{1}{\lambda \eta^{-(t-1)}} \left\| \sum_{\tau=1}^{t-W-1} \eta^{-\tau} x_\tau x_\tau^\top (\boldsymbol{\theta}_\tau - \boldsymbol{\theta}_t) \right\| \\
&\qquad \left(\left\| V_t^{-1} \right\| = \lambda_{\max}(V_t^{-1}) \leq \tfrac{1}{\lambda \eta^{-(t-1)}} \right) \\
&\leq \|x\| \frac{1}{\lambda} \sum_{\tau=1}^{t-W-1} \eta^{(t-1-\tau)} \|x_\tau\|^2 \|\boldsymbol{\theta}_\tau - \boldsymbol{\theta}_t\| \\
&\leq \|x\| \frac{2S}{\lambda} \frac{\eta^W}{1-\eta} \qquad (\|\boldsymbol{\theta}_t\| \leq S \text{ and } \|x_t\| \leq 1)
\end{aligned}$$

We conclude for any $x \in \mathbb{R}^d$

$$|x^\top (\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}}_t)| \leq \|x\| \left(\sqrt{\frac{d}{\lambda(1-\eta)}} \sum_{s=t-W}^{t-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\| + \frac{2S}{\lambda} \frac{\eta^W}{1-\eta} \right)$$

which proves that

$$\|\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}}_t\| \leq \sqrt{\frac{d}{\lambda(1-\eta)}} \sum_{s=t-W}^{t-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\| + \frac{2S}{\lambda} \frac{\eta^W}{1-\eta}$$

Comparing to the bound on $\|\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}}_t\|$ in the proof of Russac et al. [2019], there is an extra factor $\sqrt{\frac{d}{\lambda(1-\eta)}}$ that multiplies the local non-stationarity term $\sum_{s=t-W}^{t-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|$. This extra factor will consequently multiply the variation budget term in the final regret as stated in the following proposition:

Proposition 10. *Under the assumption that $\sum_{t=1}^{K-1} \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t+1}\| \leq \Delta$, for any $\delta \in (0,1)$, if we set $\beta = \sqrt{\lambda} S + \sigma \sqrt{2\log(1/\delta) + d\log(1 + \frac{1}{\lambda d(1-\gamma)})}$ in the algorithm 1* D-LinUCB *of Russac et al. [2019], then with probability $1-\delta$, for any $W > 0$ the dynamic regret of* D-LinUCB *is at most*

$$\mathcal{O}\left(\sqrt{\frac{d}{\lambda(1-\eta)}} \Delta W + \frac{S}{\lambda} \frac{\eta^W}{1-\eta} K + \beta \sqrt{dK} \sqrt{K \log(1/\eta) + \log(1 + \frac{1}{d\lambda(1-\eta)})} \right)$$

Proposition 11. *Under the same assumption as 10 If we set $\log(1/\eta) = d^{-1/4}\Delta^{1/2}K^{-1/2}$, $W = \frac{\log(K/(1-\eta))}{\log(1/\eta)}$ and $\lambda = 1$; for any $\delta \in (0,1)$; we have that with probability $1-\delta$, the dynamic regret of* D-LinUCB *is at most $\widetilde{\mathcal{O}}(d^{7/8}\Delta^{1/4}K^{3/4})$.*

Proof. With the choice $\log(1/\eta) = d^{-1/4}\Delta^{1/2}K^{-1/2}$ and $W = \frac{\log(K/(1-\eta))}{\log(1/\eta)}$; we have $\frac{\eta^W}{1-\eta}K = 1$, $\eta = \exp(-\left(\frac{\Delta}{K}\right)^{1/2})\underset{K\to\infty}{\sim}1 - d^{-1/4}\Delta^{1/2}K^{-1/2}$ so that $\sqrt{\frac{d}{\lambda(1-\eta)}}\Delta W \sim \sqrt{d}\Delta\log(K/(1-\eta))\left(d^{1/4}\Delta^{-1/2}K^{1/2}\right)^{3/2} = \tilde{\mathcal{O}}(d^{7/8}\Delta^{1/4}K^{3/4})$ and $\beta\sqrt{dK}\sqrt{K\log(1/\eta) + \log(1+\frac{1}{d\lambda(1-\eta)})} = \tilde{\mathcal{O}}(dK\left(d^{-1/4}\Delta^{1/2}K^{-1/2}\right)^{1/2}) = \tilde{\mathcal{O}}(d^{7/8}\Delta^{1/4}K^{3/4})$. □

D.3 Regret Analysis of opt-wlsvi and Proof Outline

D.3.1 Single Step Error Decomposition

In this section, we analyse the one-step error decomposition of the difference between the estimates $Q_{t,h}$ and $Q^\pi_{t,h}$ of a given policy π. To do that, we use the weighted MDP $(\mathcal{S}, \mathcal{A}, \bar{P}, \bar{r})$ to isolate the bias term. The decomposition contains four parts: the reward bias and variance, the transition bias and variance, and the difference in value functions at step $h+1$. It can be written as:

$$\begin{aligned}\phi(s,a)^\top \mathbf{w}_{t,h} - Q^\pi_{t,h}(s,a) = &\underbrace{(\bar{r}_{t,h} - r_{t,h})(s,a)}_{\text{reward bias}} + \underbrace{(\widehat{r}_{t,h} - \bar{r}_{t,h})(s,a)}_{\text{reward variance}} \\ &+ \underbrace{[(\bar{P}_{t,h} - P_{t,h})V^\pi_{t,h+1}](s,a)}_{\text{transition bias}} + \underbrace{[(\widehat{P}_{t,h} - \bar{P}_{t,h})V_{t,h}](s,a)}_{\text{transition variance}} \\ &+ \underbrace{[\bar{P}_{t,h}(V_{t,h+1} - V^\pi_{t,h+1})](s,a)}_{\text{difference in value functions of next step}} .\end{aligned}$$

The reward and transition bias terms are controlled by Lemma 9 using the fact that $\|V^\pi_{t,h}\|_\infty \le H$. The difference in value-functions at step $h+1$ can be rewritten as $[P_{t,h}(V_{t,h+1} - V^\pi_{t,h+1})](s,a) + [(\bar{P}_{t,h} - P_{t,h})(V_{t,h+1} - V^\pi_{t,h+1})](s,a)$. We control the second term by applying again Lemma 9 since $\|V_{t,h+1} - V^\pi_{t,h+1}\|_\infty \le H$.

It remains now the two variance terms. The reward variance is easy to control and it reduces simply to the bias due to the regularization as we assume that r is a deterministic function. Note that the assumption of deterministic reward is not a limiting assumption since the contribution of a stochastic reward in the final regret has lower order term than the contribution of a stochastic transition. We have, using the Cauchy-Shwartz inequality and $\left\|\widetilde{\Sigma}^{-1}_{t,h}\right\| \le \frac{1}{\lambda\eta^{-2(t-1)}}$:

$$
\begin{aligned}
|(\widehat{r}_{t,h} - \bar{r}_{t,h})(s,a)| &= \lambda \eta^{-(t-1)} |\boldsymbol{\phi}(s,a)^\top \Sigma_{t,h}^{-1} \boldsymbol{\theta}_{t,h}| \\
&\leq \sqrt{d\lambda}\, \|\boldsymbol{\phi}(s,a)\|_{\Sigma_{t,h}^{-1}\widetilde{\Sigma}_{t,h}\Sigma_{t,h}^{-1}}\,.
\end{aligned}
$$

Controlloing the transition variance is more involved, and we differ the analysis to the next section. If we define $\texttt{bias} \triangleq \texttt{bias}_r + \texttt{bias}_P$ the total non-stationarity bias of the MDP, we can summarize the one-step analysis as follows:

$$
\begin{aligned}
\boldsymbol{\phi}(s,a)^\top \mathbf{w}_{t,h} - Q_{t,h}^{\pi}(s,a) \leq & 2H\texttt{bias}(t,h) + [P_{t,h}(V_{t,h+1} - V_{t,h+1}^{\pi})](s,a) \qquad \text{(D.3.1)} \\
& + \sqrt{d\lambda}\, \|\boldsymbol{\phi}(s,a)\|_{\Sigma_{t,h}^{-1}\widetilde{\Sigma}_{t,h}\Sigma_{t,h}^{-1}} + [(\widehat{P}_{t,h} - \bar{P}_{t,h})V_{t,h}](s,a)
\end{aligned}
$$

D.3.2 High Probability Bound on the Transition Variance

In this section, we will establish a high probability bound on the term $(\widehat{P}_{t,h} - \bar{P}_{t,h})V_{t,h}$. From the definitions of $\widehat{P}$ and $\bar{P}$ and the Cauchy-Schwartz inequality, we have

$$
[(\widehat{P}_{t,h} - \bar{P}_{t,h})V_{t,h}](s,a) \leq \left(\left\| \sum_{\tau=1}^{t-1} \eta^{-\tau} \boldsymbol{\phi}_{\tau,h} \epsilon_{\tau,h} \right\|_{\widetilde{\Sigma}_{t,h}^{-1}} + H\sqrt{d\lambda} \right) \|\boldsymbol{\phi}(s,a)\|_{\Sigma_{t,h}^{-1}\widetilde{\Sigma}_{t,h}\Sigma_{t,h}^{-1}}\,,
$$

where $\epsilon_{\tau,h} = V_{t,h+1}(s_{\tau,h+1}) - [P_{t,h}V_{t,h+1}](s_{\tau,h}, a_{\tau,h})$. If $V_{t,h+1}$ was a fixed function, $\epsilon_{\tau,h}$ would be zero-mean conditioned on the history of transitions up to step h at episode τ and we would use the concentration of weighted self-normalized processes [Russac et al., 2019] to get a high probability bound on $\left\| \sum_{\tau=1}^{t-1} \eta^{-\tau} \boldsymbol{\phi}_{\tau,h} \epsilon_{\tau,h} \right\|_{\widetilde{\Sigma}_{t,h}^{-1}}$. However, as $V_{t,h+1}$ is estimated from past transitions and thus depends on the latter in a non-trivial way, we will show a concentration bound that holds uniformly for all possible value functions generated by the algorithm. We proceed first by establishing the boundness of iterates in the next Lemma.

Lemma 18 (Boundness of iterates). *For any $(t,h) \in [K] \times [H]$, the weight $\mathbf{w}_{t,h}$ and the matrix $\Sigma_t^{-1}\widetilde{\Sigma}_t\Sigma_t^{-1}$ in Algorithm 6 satisfies:*

$$
\|\mathbf{w}_{t,h}\| \leq 2H\sqrt{\frac{d(1-\eta^{t-1})}{\lambda(1-\eta)}} \text{ and } \left\|\Sigma_t^{-1}\widetilde{\Sigma}_t\Sigma_t^{-1}\right\| \leq \frac{1}{\lambda}
$$

Any value function estimate produced by Algorithm 6 could be written in the following form

$$
V^{\mathbf{w},\mathbf{A}}(\cdot) = \min\{\max_{a \in \mathcal{A}} \{\mathbf{w}^\top \boldsymbol{\phi}(\cdot,a) + \sqrt{\boldsymbol{\phi}(\cdot,a)^\top \mathbf{A} \boldsymbol{\phi}(\cdot,a)}\}, H\}
$$

where $\mathbf{w} \in \mathbb{R}^d$ and $\mathbf{A} \in \mathbb{R}^{d \times d}$ is a symmetric definite positive matrix that are in

$$\mathcal{G} = \left\{ \mathbf{w}, \mathbf{A} : \|\mathbf{w}\| \leq 2H\sqrt{\frac{d}{\lambda(1-\eta)}} \text{ and } \|\mathbf{A}\|_{\mathrm{F}} \leq \frac{\sqrt{d}\beta^2}{\lambda} \right\}$$

The ϵ-covering number of $\mathcal{G}$, identified as Euclidean ball in $\mathbb{R}^{d+d^2}$ of radius $2H\sqrt{\frac{d}{\lambda(1-\eta)}} + \frac{\beta^2\sqrt{d}}{\lambda}$, is bounded by $\left(3\left(2H\sqrt{\frac{d}{\lambda(1-\eta)}} + \frac{\beta^2\sqrt{d}}{\lambda}\right)/\epsilon\right)^{d+d^2}$. The latter number is exponential in the dimension d but only the square root of its logarithm, which is linear in d, will contribute to the bound as we will show next.

By applying the concentration of weighted self-normalized processes [Russac et al., 2019] and using a union bound argument over an ϵ-net of $\mathcal{G}$ with an appropriate value of ϵ, we obtain the desired high probability bound stated in the following Lemma

Lemma 19. *For any $\delta \in (0,1)$, with probability at least $1-\delta/2$, we have for all $(t,h) \in [K] \times [H]$,*

$$\left\| \sum_{\tau=1}^{t-1} \eta^{-\tau} \phi_{\tau,h} \epsilon_{\tau,h} \right\|_{\widetilde{\Sigma}_{t,h}^{-1}} \leq CdH\sqrt{\log\left(\frac{dH\beta}{\lambda(1-\eta)} \cdot \frac{2}{\delta}\right)}$$

where $C > 0$ is an absolute constant.

Finally, by combining the single error decomposition in Equation (D.3.1) and the transition concentration in Lemma 19 with an appropriate choice of β, we obtain the following high probability single-step bound.

Lemma 20 (Key lemma)**.** *There exists an absolute value c such that $\beta = cdH\sqrt{\iota}$ where $\iota = \log\left(\frac{2dH}{(1-\eta)\delta}\right)$, $\lambda = 1$ and for any fixed policy π, we have with probability at least $1-\delta/2$ for all $(s,a,h,t) \in \mathcal{S} \times \mathcal{A} \times [H] \times [K]$,*

$$\begin{aligned} &\left| \phi(s,a)^\top \mathbf{w}_{t,h} - Q_{t,h}^{\pi}(s,a) - [P_{t,h}(V_{t,h+1} - V_{t,h+1}^{\pi})](s,a) \right| \\ &\quad \leq 2H\,\mathtt{bias}(t,h) + \beta \left\|\phi(s,a)\right\|_{\Sigma_{t,h}^{-1}\widetilde{\Sigma}_{t,h}\Sigma_{t,h}^{-1}}. \end{aligned}$$

D.3.3 Optimism

Now, we show that the true value functions can be upper bounded by the value functions computed by OPT-WLSVI plus a bias term. In fact, unlike the stationary case, we act optimistically with respect to the weighted average MDP. To prove this, we use the key Lemma 20 in the previous section and we proceed by induction argument over steps $h \in [H]$.

Lemma 21 (Optimism). *For all* $(s, a, t, h) \in \mathcal{S} \times \mathcal{A} \times [K] \times [H]$*, we have with probability at least* $1 - \delta/2$

$$Q_{t,h}(s,a) + 2H \sum_{h'=h}^{H} \texttt{bias}(t,h) \geq Q^{\star}_{t,h}(s,a) \tag{D.3.2}$$

D.3.4 Final Regret Analysis

Now, having the results provided in previous sections at hand, we turn to proving the regret bound of our algorithm. Let π_t the policy executed by the algorithm in step h for H steps to reach the end of the episode. If we define $\delta_{t,h} \triangleq V_{t,h}(s_{t,1}) - V^{\pi_t}_{t,h}(s_{t,h})$, a straightforward application of Lemma 21 is that the regret is upper bounded by the sum of $\delta_{t,h}$ and bias terms with probability at least $1 - \delta/2$ i.e

$$\textsc{Regret}(K) \underbrace{\leq}_{\text{by optimism}} \sum_{t=1}^{K} \delta_{t,h} + 2H \sum_{t=1}^{K} \sum_{h=1}^{H} \texttt{bias}(t,h). \tag{D.3.3}$$

The policy π_t is the greedy policy with respect to $Q_{t,h}$, and $a_{t,h} = \pi_t(s_t, h) = \arg\max_{a \in \mathcal{A}} Q_{t,h}(s_{t,h}, a)$. Therefore, we have $\delta_{t,h} = Q_{t,h}(s_{t,h}, a_{t,h}) - Q^{\pi_t}_{t,h}(s_{t,h}, a_{t,h})$. Using the definition of $Q_{t,h}$ and the key Lemma 20, we obtain with probability at least $1 - \delta/2$,

$$\begin{aligned}
\delta_{t,h} &\leq [P_{t,h}(V_{t,h+1} - V^{\pi_t}_{t,h+1})](s_{t,h}, a_{t,h}) + 2\beta \left\|\phi_{t,h}\right\|_{\Sigma^{-1}_{t,h}\widetilde{\Sigma}_{t,h}\Sigma^{-1}_{t,h}} + 2H\texttt{bias}(t,h) \\
&= \delta_{t,h+1} + \xi_{t,h+1} + 2\beta \left\|\phi_{t,h}\right\|_{\Sigma^{-1}_{t,h}\widetilde{\Sigma}_{t,h}\Sigma^{-1}_{t,h}} + 2H\texttt{bias}(t,h)
\end{aligned}$$

where we define $\xi_{t,h+1} = [P_{t,h}(V_{t,h+1} - V^{\pi_t}_{t,h+1})](s_{t,h}, a_{t,h}) - (V_{t,h+1} - V^{\pi_t}_{t,h+1})(s_{t,h+1})$. Unrolling the last inequality H times, we obtain

$$\delta_{t,1} \leq \sum_{h=1}^{H} \xi_{t,h} + 2\beta \sum_{h=1}^{H} \left\|\phi_{t,h}\right\|_{\Sigma^{-1}_{t,h}\widetilde{\Sigma}_{t,h}\Sigma^{-1}_{t,h}} + 2H \sum_{h=1}^{H} \texttt{bias}(t,h) \tag{D.3.4}$$

Hence, by combining Equations (D.3.3) and (D.3.4), we obtain with probability at least $1 - \delta/2$,

$$\begin{aligned}
\textsc{Regret}(K) \leq \underbrace{\sum_{t=1}^{K} \sum_{h=1}^{H} \xi_{t,h}}_{(A)} + 2\beta \underbrace{\sum_{t=1}^{K} \sum_{h=1}^{H} \left\|\phi_{t,h}\right\|_{\Sigma^{-1}_{t,h}\widetilde{\Sigma}_{t,h}\Sigma^{-1}_{t,h}}}_{(B)} \\
+ 4H \underbrace{\sum_{t=1}^{K} \sum_{h=1}^{H} \texttt{bias}(t,h)}_{(C)}.
\end{aligned} \tag{D.3.5}$$

Now, we proceed to upper bound the different terms in the RHS of Equation (D.3.5).

Term (A): The computation of $V_{t,h}$ is independent from $(s_{t,h}, a_{t,h})$, therefore, $\{\xi_{t,h}\}$ is $2H$-bounded martingale difference sequence. Therefore, by Azuma-Hoeffding, we have for all $t > 0$, $P\left(\left|\sum_{t=1}^{K}\sum_{h=1}^{H}\xi_{t,h}\right| \geq t\right) \leq 2\exp\left(\frac{-t^2}{16H^3K}\right)$. Then, $P\left(\left|\sum_{t=1}^{K}\sum_{h=1}^{H}\xi_{t,h}\right| \geq 4\sqrt{H^3K\log(4/\delta)}\right) \leq \delta/2$. Therefore, with probability at least $1-\delta/2$, we have

$$\left|\sum_{t=1}^{K}\sum_{h=1}^{H}\xi_{t,h}\right| \leq \mathcal{O}(H^{3/2}\sqrt{Kt}) \tag{D.3.6}$$

Term (B): By application of Cauchy-Schwartz, we obtain

$$\sum_{t=1}^{K}\sum_{h=1}^{H}\|\phi_{t,h}\|_{\Sigma_{t,h}^{-1}\widetilde{\Sigma}_{t,h}\Sigma_{t,h}^{-1}} \leq \sqrt{K}\sum_{h=1}^{H}\sqrt{\sum_{t=1}^{K}\|\phi_{t,h}\|^2_{\Sigma_{t,h}^{-1}\widetilde{\Sigma}_{t,h}\Sigma_{t,h}^{-1}}}.$$

From Lemma 18, we have $\left\|\Sigma_t^{-1}\widetilde{\Sigma}_t\Sigma_t^{-1}\right\| \leq \frac{1}{\lambda}$, then, $\|\phi_{t,h}\|_{\Sigma_{t,h}^{-1}\widetilde{\Sigma}_{t,h}\Sigma_{t,h}^{-1}} \leq \frac{1}{\sqrt{\lambda}}\|\phi_{t,h}\| = \|\phi_{t,h}\| \leq 1$. So, we can use the bound on the sum of the squared norm of the features provided in proposition 4 of Russac et al. [2019] to obtain

$$\sum_{t=1}^{K}\sum_{h=1}^{H}\|\phi_{t,h}\|_{\Sigma_{t,h}^{-1}\widetilde{\Sigma}_{t,h}\Sigma_{t,h}^{-1}} \leq H\sqrt{K}\sqrt{2dK\log(1/\eta) + 2d\log\left(1+\frac{1}{d\lambda(1-\eta)}\right)}. \tag{D.3.7}$$

Term (C): We control the bias term using the MDP variation budget as follows.

$$\begin{aligned}
&\sum_{t=1}^{K}\sum_{h=1}^{H}\texttt{bias}(t,h) \leq \frac{4HK\sqrt{d}}{\lambda}\frac{\eta^W}{1-\eta} + \sqrt{\frac{d}{\lambda(1-\eta)}}\cdot \\
&\sum_{t=1}^{K}\sum_{h=1}^{H}\sum_{s=t-W}^{t-1}\|\boldsymbol{\theta}_{s,h} - \boldsymbol{\theta}_{s+1,h}\| + \|\boldsymbol{\mu}_{s,h}(\mathcal{S}) - \boldsymbol{\mu}_{s+1,h}(\mathcal{S})\| \\
&\leq \frac{4HK\sqrt{d}}{\lambda}\frac{\eta^W}{1-\eta} + \sqrt{\frac{d}{\lambda(1-\eta)}}W\Delta.
\end{aligned} \tag{D.3.8}$$

Finally, the desired regret bound in Theorem 4 is obtained by combining Equations (D.3.5), (D.3.6), (D.3.7) and (D.3.8).

D.4 Missing Proofs of Regret Analysis of OPT-WLSVI

D.4.1 Linearity of Q-values: Lemma 8

Proof. The definition of non-stationary linear MDP from Assumption 6 together with the Bellman equation gives:

$$\begin{aligned} Q^{\pi}_{t,h} &= r_{t,h}(s,a) + [\mathbb{P}_{t,h} V^{\pi}_{t,h+1}](s,a) \\ &= \boldsymbol{\phi}(s,a)^{\top} \boldsymbol{\theta}_{t,h} + \int_{s'} \boldsymbol{\phi}(s,a)^{\top} V^{\pi}_{t,h+1}(s') d\boldsymbol{\mu}_{t,h}(s') \\ &= \boldsymbol{\phi}(s,a)^{\top} \left(\boldsymbol{\theta}_{t,h} + \int_{s'} V^{\pi}_{t,h+1}(s') d\boldsymbol{\mu}_{t,h}(s') \right) \end{aligned}$$

We define $\mathbf{w}^{\pi}_{t,h}$ to be the term inside the parentheses. □

D.4.2 Non-Stationarity Bias

Proof of Lemma 9

Reward Bias:

$$\begin{aligned} & |r_{t,h}(s,a) - \bar{r}_{t,h}(s,a)| \\ & \leq \left| \boldsymbol{\phi}(s,a)^{\top} \left(\boldsymbol{\theta}_{t,h} - \Sigma^{-1}_{t,h} \left(\sum_{\tau=1}^{t-1} \eta^{-\tau} \boldsymbol{\phi}_{\tau,h} \boldsymbol{\phi}^{\top}_{\tau,h} \boldsymbol{\theta}_{\tau,h} + \lambda \eta^{-(t-1)} \boldsymbol{\theta}_{t,h} \right) \right) \right| \\ & = \left| \boldsymbol{\phi}(s,a)^{\top} \sum_{\tau=1}^{t-1} \Sigma^{-1}_{t,h} \eta^{-\tau} \boldsymbol{\phi}_{\tau,h} \boldsymbol{\phi}^{\top}_{\tau,h} (\boldsymbol{\theta}_{t,h} - \boldsymbol{\theta}_{\tau,h}) \right| \\ & \leq \underbrace{\left| \boldsymbol{\phi}(s,a)^{\top} \sum_{\tau=t-W}^{t-1} \Sigma^{-1}_{t,h} \eta^{-\tau} \boldsymbol{\phi}_{\tau,h} \boldsymbol{\phi}^{\top}_{\tau,h} (\boldsymbol{\theta}_{t,h} - \boldsymbol{\theta}_{\tau,h}) \right|}_{(\star)} + \underbrace{\left| \boldsymbol{\phi}(s,a)^{\top} \sum_{\tau=1}^{t-W-1} \Sigma^{-1}_{t,h} \eta^{-\tau} \boldsymbol{\phi}_{\tau,h} \boldsymbol{\phi}^{\top}_{\tau,h} (\boldsymbol{\theta}_{t,h} - \boldsymbol{\theta}_{\tau,h}) \right|}_{(\star\star)} \end{aligned}$$

Bound on $(\star)$:

$$
\begin{aligned}
&\left|\phi(s,a)^\top \sum_{\tau=t-W}^{t-1} \Sigma_{t,h}^{-1}\eta^{-\tau}\phi_{\tau,h}\phi_{\tau,h}^\top(\boldsymbol{\theta}_{t,h}-\boldsymbol{\theta}_{\tau,h})\right| \\
&= \left|\sum_{\tau=t-W}^{t-1} \eta^{-\tau}\phi(s,a)^\top\Sigma_{t,h}^{-1}\phi_{\tau,h}\phi_{\tau,h}^\top(\boldsymbol{\theta}_{t,h}-\boldsymbol{\theta}_{\tau,h})\right| \\
&\le \sum_{\tau=t-W}^{t-1} \eta^{-\tau}\left|\phi(s,a)^\top\Sigma_{t,h}^{-1}\phi_{\tau,h}\right|\cdot\left|\phi_{\tau,h}^\top(\boldsymbol{\theta}_{t,h}-\boldsymbol{\theta}_{\tau,h})\right| \\
&\le \sum_{\tau=t-W}^{t-1} \eta^{-\tau}\left|\phi(s,a)^\top\Sigma_{t,h}^{-1}\phi_{\tau,h}\right|\|\phi_{\tau,h}\|\,\|\boldsymbol{\theta}_{t,h}-\boldsymbol{\theta}_{\tau,h}\| \\
&\le \sum_{\tau=t-W}^{t-1} \eta^{-\tau}\left|\phi(s,a)^\top\Sigma_{t,h}^{-1}\phi_{\tau,h}\right|\|\boldsymbol{\theta}_{t,h}-\boldsymbol{\theta}_{\tau,h}\| && (\|\phi_{\tau,h}\|\le 1) \\
&= \sum_{\tau=t-W}^{t-1} \eta^{-\tau}\left|\phi(s,a)^\top\Sigma_{t,h}^{-1}\phi_{\tau,h}\right|\left\|\sum_{s=\tau}^{t-1}\boldsymbol{\theta}_{s,h}-\boldsymbol{\theta}_{s+1,h}\right\| \\
&\le \sum_{\tau=t-W}^{t-1} \eta^{-\tau}\left|\phi(s,a)^\top\Sigma_{t,h}^{-1}\phi_{\tau,h}\right|\sum_{s=\tau}^{t-1}\|\boldsymbol{\theta}_{s,h}-\boldsymbol{\theta}_{s+1,h}\| \\
&\le \sum_{s=t-W}^{t-1}\sum_{\tau=t-W}^{s} \eta^{-\tau}\left|\phi(s,a)^\top\Sigma_{t,h}^{-1}\phi_{\tau,h}\right|\|\boldsymbol{\theta}_{s,h}-\boldsymbol{\theta}_{s+1,h}\| \\
&\qquad\qquad (\textstyle\sum_{\tau=t-W}^{t-1}\sum_{s=\tau}^{t-1}=\sum_{s=t-W}^{t-1}\sum_{\tau=t-W}^{s}) \\
&\le \sum_{s=t-W}^{t-1}\sqrt{\left[\sum_{\tau=t-W}^{s}\eta^{-\tau}\phi(s,a)^\top\Sigma_{t,h}^{-1}\phi(s,a)\right]\cdot\left[\sum_{\tau=t-W}^{s}\eta^{-\tau}\phi_{\tau,h}^\top\Sigma_{t,h}^{-1}\phi_{\tau,h}\right]} \\
&\quad\cdot\|\boldsymbol{\theta}_{s,h}-\boldsymbol{\theta}_{s+1,h}\| && \text{(Cauchy-Schwartz)} \\
&\le \sum_{s=t-W}^{t-1}\sqrt{\left[\sum_{\tau=t-W}^{s}\eta^{-\tau}\phi(s,a)^\top\Sigma_{t,h}^{-1}\phi(s,a)\right]\cdot\sqrt{d}}\cdot\|\boldsymbol{\theta}_{s,h}-\boldsymbol{\theta}_{s+1,h}\| \\
&\qquad\qquad \text{(by Lemma 22)} \\
&\le \sum_{s=t-W}^{t-1}\sqrt{d}\sum_{s=t-W}^{t-1}\sqrt{\frac{\sum_{\tau=t-W}^{t-1}\eta^{-\tau}}{\lambda\eta^{-(t-1)}}}\|\boldsymbol{\theta}_{s,h}-\boldsymbol{\theta}_{s+1,h}\| \\
&\qquad\qquad (\|\phi(s,a)\|\le 1 \text{ and } \lambda_{\max}(\Sigma_{t,h}^{-1})\le\tfrac{1}{\lambda\eta^{-(t-1)}}) \\
&\le \sqrt{\frac{d}{\lambda(1-\eta)}}\sum_{s=t-W}^{t-1}\|\boldsymbol{\theta}_{s,h}-\boldsymbol{\theta}_{s+1,h}\|
\end{aligned}
$$

Bound on $(\star\star)$:

$$\begin{aligned}
&\left|\phi(s,a)^\top \sum_{\tau=1}^{t-W-1} \Sigma_{t,h}^{-1}\eta^{-\tau}\phi_{\tau,h}\phi_{\tau,h}^\top(\boldsymbol{\theta}_{t,h}-\boldsymbol{\theta}_{\tau,h})\right| \\
&\leq \frac{1}{\lambda}\|\phi(s,a)\| \sum_{\tau=1}^{t-W-1} \eta^{t-\tau-1}\|\phi_{\tau,h}\|\cdot|\phi_{\tau,h}^\top(\boldsymbol{\theta}_{t,h}-\boldsymbol{\theta}_{\tau,h})| \quad (\lambda_{\max}(\Sigma_{t,h}^{-1}) \leq \tfrac{1}{\lambda\eta^{-(t-1)}}) \\
&\leq \frac{1}{\lambda}\|\phi(s,a)\| \sum_{\tau=1}^{t-W-1} \eta^{t-\tau-1}\|\phi_{\tau,h}\|^2\|\boldsymbol{\theta}_{t,h}-\boldsymbol{\theta}_{\tau,h}\| \\
&\leq \frac{2\sqrt{d}}{\lambda}\frac{\eta^W}{1-\eta} \qquad (\phi(s,a)\leq 1 \text{ and } \|\boldsymbol{\theta}_{t,h}\|\leq\sqrt{d})
\end{aligned}$$

Transition Bias: $\forall f : \mathcal{S} \to \mathbb{R}$ such that $\|f\|_\infty < \infty$ (real-valued bounded function), similarly to what we have done for the reward function, we obtain

$$\begin{aligned}
|[(\mathbb{P}_{t,h}-\bar{\mathbb{P}}_{t,h})f](s,a)| &\leq \left\|\Sigma_{t,h}^{-1}\sum_{\tau=1}^{t-1}\eta^{-\tau}\phi_{\tau,h}\phi_{\tau,h}^\top\int f(s')(d\boldsymbol{\mu}_{t,h}(s')-d\boldsymbol{\mu}_{\tau,h}(s'))\right\| \\
&\qquad (\|\phi(s,a)\|\leq 1) \\
&\leq \sqrt{\frac{d}{\lambda(1-\eta)}}\sum_{s=t-W}^{t-1}\left\|\int f(s')(d\boldsymbol{\mu}_{s,h}(s')-d\boldsymbol{\mu}_{s+1,h}(s'))\right\| \\
&\quad + \frac{1}{\lambda}\sum_{\tau=1}^{t-W-1}\eta^{t-\tau-1}\left\|\int f(s')(d\boldsymbol{\mu}_{t,h}(s')-d\boldsymbol{\mu}_{\tau,h}(s'))\right\|
\end{aligned}$$

Furthermore,

$$\begin{aligned}
\left\|\int f(s')(d\boldsymbol{\mu}_{t,h}(s')-d\boldsymbol{\mu}_{\tau,h}(s'))\right\| &= \sqrt{\sum_{l=1}^{d}\left|\int f(s')(d\boldsymbol{\mu}_{t,h}^{(l)}(s')-d\boldsymbol{\mu}_{\tau,h}^{(l)}(s'))\right|^2} \\
&\leq \|f\|_\infty\sqrt{\sum_{l=1}^{d}|\boldsymbol{\mu}_{t,h}^{(l)}(\mathcal{S})-\boldsymbol{\mu}_{\tau,h}^{(l)}(\mathcal{S}))|^2} \\
&= \|f\|_\infty\|\boldsymbol{\mu}_{s,h}(\mathcal{S})-\boldsymbol{\mu}_{s+1,h}(\mathcal{S})\| \\
&\leq 2\sqrt{d}\|f\|_\infty
\end{aligned}$$

Therefore,

$$|[(\mathbb{P}_{t,h}-\bar{\mathbb{P}}_{t,h})f](s,a)| \leq \|f\|_\infty\left(\sqrt{\frac{d}{\lambda(1-\eta)}}\sum_{s=t-W}^{t-1}\|(\boldsymbol{\mu}_{s,h}(\mathcal{S})-\boldsymbol{\mu}_{s+1,h}(\mathcal{S}))\| + \frac{2\sqrt{d}}{\lambda}\frac{\eta^W}{1-\eta}\right)$$

D.4.3 Single Step Error Decomposition

We provide here the full derivation of the single-error decomposition. We have for all $(t,h) \in [K] \times [H]$

$$\begin{aligned}\phi(s,a)^\top \mathbf{w}_{t,h} - Q^\pi_{t,h}(s,a) = \quad & \underbrace{(\bar{r}_{t,h} - r_{t,h})(s,a)}_{\text{reward bias}} + \underbrace{(\widehat{r}_{t,h} - \bar{r}_{t,h})(s,a)}_{\text{reward variance}} + \\ & \underbrace{[(\bar{P}_{t,h} - P_{t,h})V^\pi_{t,h+1}](s,a)}_{\text{transition bias}} + \underbrace{[(\widehat{P}_{t,h} - \bar{P}_{t,h})V_{t,h+1}](s,a)}_{\text{transition variance}} + \\ & \underbrace{[\bar{P}_{t,h}(V_{t,h+1} - V^\pi_{t,h+1})](s,a)}_{\text{difference in value functions of next step}} \quad .\end{aligned}$$

Reward & transition bias: Thanks to Lemma 9, we have

$$\begin{aligned}|\bar{r}_{t,h}(s,a) - r_{t,h}(s,a)| &\le \texttt{bias}_r(t,h), \\ \left|[(\bar{P}_{t,h} - P_{t,h})V^\pi_{t,h+1}](s,a)\right| &\le \texttt{bias}_P(t,h). \qquad (\|V^\pi_{t,h+1}\|_\infty \le H)\end{aligned}$$

Difference in value functions of next step:

$$\begin{aligned}[\bar{P}_{t,h}(V_{t,h+1} - V^\pi_{t,h+1})](s,a) &= [P_{t,h}(V_{t,h+1} - V^\pi_{t,h+1})](s,a) + [(\bar{P}_{t,h} - P_{t,h})(V_{t,h+1} - V^\pi_{t,h+1})](s,a) \\ &\le [P_{t,h}(V_{t,h+1} - V^\pi_{t,h+1})](s,a) + \texttt{bias}_P(s,a). \\ &\qquad (\|V_{t,h+1} - V^\pi_{t,h+1}\|_\infty \le H)\end{aligned}$$

Reward variance: The reward variance here reduces simply to the bias due to the regularization as we assume that r is a deterministic function.

$$\begin{aligned}\left|(\widehat{r}_{t,h} - \bar{r}_{t,h})(s,a)\right| &= \lambda\eta^{-(t-1)}|\langle \phi(s,a), \Sigma^{-1}_{t,h}\boldsymbol{\theta}_{t,h}\rangle| \\ &\le \lambda\eta^{-(t-1)} \|\phi(s,a)\|_{\Sigma^{-1}_{t,h}\widetilde{\Sigma}_{t,h}\Sigma^{-1}_{t,h}} \left\|\Sigma^{-1}_{t,h}\boldsymbol{\theta}_{t,h}\right\|_{\Sigma_{t,h}\widetilde{\Sigma}^{-1}_{t,h}\Sigma_{t,h}} \\ &= \lambda\eta^{-(t-1)} \|\phi(s,a)\|_{\Sigma^{-1}_{t,h}\widetilde{\Sigma}_{t,h}\Sigma^{-1}_{t,h}} \|\boldsymbol{\theta}_{t,h}\|_{\widetilde{\Sigma}^{-1}_{t,h}} \\ &\le \lambda\eta^{-(t-1)} \|\phi(s,a)\|_{\Sigma^{-1}_{t,h}\widetilde{\Sigma}_{t,h}\Sigma^{-1}_{t,h}} \sqrt{\left\|\widetilde{\Sigma}^{-1}_{t,h}\right\|} \|\boldsymbol{\theta}_{t,h}\| \\ &\le \sqrt{d\lambda} \|\phi(s,a)\|_{\Sigma^{-1}_{t,h}\widetilde{\Sigma}_{t,h}\Sigma^{-1}_{t,h}}\end{aligned}$$

The last step follows from $\|\boldsymbol{\theta}_{t,h}\| \le \sqrt{d}$ (Assumption 6) and $\left\|\widetilde{\Sigma}^{-1}_{t,h}\right\| \le \frac{1}{\lambda\eta^{-2(t-1)}}$.

If we define $\texttt{bias} \triangleq \texttt{bias}_r + 2 \cdot \texttt{bias}_P$ the total non-stationarity bias of the MDP, we can summarize the one-step analysis as follows:

$$\phi(s,a)^\top \mathbf{w}_{t,h} - Q^\pi_{t,h}(s,a) \leq \quad \texttt{bias}(t,h) + [\mathbb{P}_{t,h}(V_{t,h+1} - V^\pi_{t,h+1})](s,a) + \sqrt{d\lambda}\,\|\phi(s,a)\|_{\Sigma^{-1}_{t,h}\widetilde{\Sigma}_{t,h}\Sigma^{-1}_{t,h}} + [(\widehat{P}_{t,h} - \bar{P}_{t,h})V_{t,h}](s,a)$$

D.4.4 Boundness of iterates

We will start with the following elementary lemma:

Lemma 22. *Let* $\Sigma_t = \sum_{\tau=1}^{t-1} \eta^{-\tau}\phi_\tau\phi_\tau^\top + \lambda\eta^{-(t-1)}\mathbf{I}$ *where* $\phi_\tau \in \mathbb{R}^d$ *and* $\lambda > 0, \eta \in (0,1)$. *Then:*

$$\sum_{\tau=1}^{t-1} \eta^{-\tau}\phi_\tau^\top\Sigma_t^{-1}\phi_\tau \leq d$$

Proof. We have $\sum_{\tau=1}^{t-1}\eta^{-\tau}\phi_\tau^\top\Sigma_t^{-1}\phi_\tau = \sum_{\tau=1}^{t-1}\mathrm{tr}\left(\eta^{-\tau}\phi_\tau^\top\Sigma_t^{-1}\phi_\tau\right) = \mathrm{tr}\left(\Sigma_t^{-1}\sum_{\tau=1}^{t-1}\eta^{-\tau}\phi_\tau\phi_\tau^\top\right)$. Given the eigenvalue decomposition $\sum_{\tau=1}^{t-1}\eta^{-\tau}\phi_\tau\phi_\tau = \mathrm{diag}(\lambda_1,\ldots,\lambda_d)^\top$, we have $\Sigma_t = \mathrm{diag}(\lambda_1 + \lambda\eta^{-(t-1)},\ldots,\lambda_d + \lambda\eta^{-(t-1)})^\top$, and $\mathrm{tr}\left(\Sigma_t^{-1}\sum_{\tau=1}^{t-1}\eta^{-\tau}\phi_\tau\phi_\tau^\top\right) = \sum_{i=1}^d \frac{\lambda_j}{\lambda_j+\lambda\eta^{-(t-1)}} \leq d$ □

Proof of Lemma 18

Bound on $\|\mathbf{w}_{t,h}\|$: For any vector $v \in \mathbb{R}^d$, we have

$$|v^\top\mathbf{w}_{t,h}| = \left|v^\top\Sigma_{t,h}^{-1}\sum_{\tau=1}^{t-1}\eta^{-\tau}\phi_{\tau,h}[r_{\tau,h} + \max_a Q_{\tau,h+1}(s^{\tau,h+1},a)]\right|$$
$$\leq \sum_{\tau=1}^{t-1}\eta^{-\tau}|v^\top\Sigma_{t,h}^{-1}\phi_{\tau,h}|\cdot 2H \leq \sqrt{\left[\sum_{\tau=1}^{t-1}\eta^{-\tau}v^\top\Sigma_{t,h}^{-1}v\right]\cdot\left[\sum_{\tau=1}^{t-1}\eta^{-\tau}\phi_{\tau,h}^\top\Sigma_{t,h}^{-1}\phi_{\tau,h}\right]}\cdot 2H$$
$$\leq 2H\|v\|\sqrt{\frac{\sum_{\tau=1}^{t-1}\eta^{-\tau}}{\lambda\eta^{-(t-1)}}\cdot d} = 2H\|v\|\sqrt{\frac{d(1-\eta^{t-1})}{\lambda(1-\eta)}}$$

where the third inequality is due to Lemma 22 and the fact that the eigenvalues of $\Sigma_{t,h}^{-1}$ are upper bounded by $\frac{1}{\lambda\eta^{-(t-1)}}$. The remainder of the proof follows from the fact that $\|\mathbf{w}_{t,h}\| = \max_{v:\|v\|=1}|v^\top\mathbf{w}_{t,h}|$.

Bound on $\left\|\Sigma_t^{-1}\widetilde{\Sigma}_t\Sigma_t^{-1}\right\|$:

$$\widetilde{\Sigma}_t = \sum_{\tau=1}^{t-1}\eta^{-2\tau}\phi_\tau\phi_\tau^\top + \lambda\eta^{-2(t-1)}\mathbf{I} \leq \eta^{-(t-1)}\sum_{\tau=1}^{t-1}\eta^{-\tau}\phi_\tau\phi_\tau^\top + \lambda\eta^{-2(t-1)}\mathbf{I} = \eta^{-(t-1)}\Sigma_t \tag{D.4.1}$$

Hence,

$$\Sigma_t^{-1}\widetilde{\Sigma}_t\Sigma_t^{-1} \leq \eta^{-(t-1)}\Sigma_t^{-1}\Sigma_t\Sigma_t^{-1} = \eta^{-(t-1)}\Sigma_t^{-1}$$

and

$$\left\|\Sigma_t^{-1}\widetilde{\Sigma}_t\Sigma_t^{-1}\right\| \leq \eta^{-(t-1)}\left\|\Sigma_t^{-1}\right\| \leq \eta^{-(t-1)}\frac{1}{\lambda\eta^{-(t-1)}} = \frac{1}{\lambda}$$

D.4.5 Transition Concentration

$$\begin{aligned}
&\left|[(\widehat{P}_{t,h} - \bar{P}_{t,h})V_{t,h}](s,a)\right| \\
&\leq \left|\boldsymbol{\phi}(s,a)^\top\left(\Sigma_{t,h}^{-1}\sum_{\tau=1}^{k-1}\eta^{-\tau}\boldsymbol{\phi}_{\tau,h}(V_{t,h+1}(s_{\tau,h+1}) - [\mathbb{P}_{t,h}V_{t,h+1}](s_{\tau,h},a_{\tau,h}))\right.\right. \\
&\left.\left. - \lambda\eta^{-(t-1)}\Sigma_{t,h}^{-1}\boldsymbol{\mu}_{t,h}V_{t,h+1}\right)\right| \\
&\leq \|\boldsymbol{\phi}(s,a)\|_{\Sigma_{t,h}^{-1}\widetilde{\Sigma}_{t,h}\Sigma_{t,h}^{-1}}\left(\left\|\sum_{\tau=1}^{t-1}\eta^{-\tau}\boldsymbol{\phi}_{\tau,h}\left(V_{t,h+1}(s_{\tau,h+1}) - [\mathbb{P}_{t,h}V_{t,h+1}](s_{\tau,h},a_{\tau,h})\right)\right\|_{\widetilde{\Sigma}_{t,h}^{-1}}\right. \\
&\quad \left. + \lambda\eta^{-(t-1)}\sqrt{\left\|\widetilde{\Sigma}_{t,h}^{-1}\right\|}\,\|\boldsymbol{\mu}_{t,h}(\mathcal{S})\|\,\|V_{t,h+1}\|_\infty\right) \qquad \text{(Cauchy-Schwarz)} \\
&\leq \|\boldsymbol{\phi}(s,a)\|_{\Sigma_{t,h}^{-1}\widetilde{\Sigma}_{t,h}\Sigma_{t,h}^{-1}}\left(\left\|\sum_{\tau=1}^{t-1}\eta^{-\tau}\boldsymbol{\phi}_{\tau,h}\left(V_{t,h+1}(s_{\tau,h+1}) - [\mathbb{P}_{t,h}V_{t,h+1}](s_{\tau,h},a_{\tau,h})\right)\right\|_{\widetilde{\Sigma}_{t,h}^{-1}} + H\sqrt{d\lambda}\right)
\end{aligned} \tag{D.4.2}$$

The last step follows from $\|\boldsymbol{\mu}_{t,h}(\mathcal{S})\| \leq \sqrt{d}$ (Assumption 6) and $\left\|\widetilde{\Sigma}_{t,h}^{-1}\right\| \leq \frac{1}{\lambda\eta^{-2(t-1)}}$.

Let us now consider the following function form:

$$V^{\mathbf{w},\mathbf{A}}(\cdot) = \min\{\max_{a\in\mathcal{A}}\{\mathbf{w}^\top\boldsymbol{\phi}(\cdot,a) + \sqrt{\boldsymbol{\phi}(\cdot,a)^\top\mathbf{A}\boldsymbol{\phi}(\cdot,a)}\}, H\} \tag{D.4.3}$$

where $\mathbf{w} \in \mathbb{R}^d$ and $\mathbf{A} \in \mathbb{R}^{d\times d}$ is a symmetric definite positive matrix that are in

$$\mathcal{G} = \left\{\mathbf{w},\mathbf{A} : \|\mathbf{w}\| \leq 2H\sqrt{\frac{d}{\lambda(1-\eta)}} \text{ and } \|\mathbf{A}\|_{\mathrm{F}} \leq \frac{\sqrt{d}\beta^2}{\lambda}\right\} \tag{D.4.4}$$

In the technical Lemma 26, we prove a concentration bound that holds uniformly for any function on the form $V^{\mathbf{w},\mathbf{A}}$ where $\mathbf{w},\mathbf{A} \in \mathcal{G}$. The statement and full proof of this lemma is defered to section D.5 of the appendix. As a corollary of Lemma 26, we can prove the concentration of the transition as follows.

For any $\tau > 0, h \in [H]$, let $\mathcal{F}_{\tau,h}$ be the σ-field generated by all the random variables until episode τ, step h. $\{s_{\tau,h}\}$ defines a stochastic process on state space $\mathcal{S}$ with corresponding filtration $\{\mathcal{F}_{\tau,h}\}$. We have

$$\begin{aligned} V_{\tau,h+1}(\cdot) &= \max_{a\in\mathcal{A}}\{\min\{\mathbf{w}_{t,h+1}^\top \phi(\cdot,a) + \beta_t[\phi(\cdot,a)^\top \Sigma_{t,h}^{-1}\widetilde{\Sigma}_{t,h}\Sigma_{t,h}^{-1}\phi(\cdot,a)]^{1/2}, H\}\} \\ &= V^{\mathbf{w},\mathbf{A}}(\cdot) \end{aligned}$$

where $\mathbf{w} = \mathbf{w}_{t,h+1}$ and $\mathbf{A} = \beta_t^2\Sigma_{t,h}^{-1}\widetilde{\Sigma}_{t,h}\Sigma_{t,h}^{-1}$. We have $\|\mathbf{w}\| \le 2H\sqrt{\frac{d(1-\eta^{t-1})}{\lambda(1-\eta)}}$ and $\|\mathbf{A}\|_F \le \sqrt{d}\beta_t^2\|A\| \le \frac{\sqrt{d}\beta_t^2}{\lambda}$ by Lemma 18. Therefore $(\mathbf{w},\mathbf{A}) \in \mathcal{G}$ and we can apply Lemma 26: we have with probability at least $1-\delta/2$

$$\begin{aligned} &\left\|\sum_{\tau=1}^{t-1}\eta^{-\tau}\phi_{\tau,h}\left(V_{t,h+1}(s_{\tau,h+1}) - [\mathbb{P}_{t,h}V_{t,h+1}](s_{\tau,h},a_{\tau,h})\right)\right\|_{\widetilde{\Sigma}_{t,h}^{-1}} \\ &\le CdH\sqrt{\log\left(dH\beta\frac{1}{\lambda(1-\eta)}\right) + \log(2/\delta)} \end{aligned} \tag{D.4.5}$$

Combining Equation (D.4.2) and (D.4.5), we obtain that with probability at least $1-\delta/2$:

$$\begin{aligned} &\left|[(\widehat{P}_{t,h} - \bar{P}_{t,h})V_{t,h}](s,a)\right| \\ &\le \|\phi(s,a)\|_{\Sigma_{t,h}^{-1}\widetilde{\Sigma}_{t,h}\Sigma_{t,h}^{-1}}\left(CdH\sqrt{\log\left(dH\beta\frac{1}{\lambda(1-\eta)}\right) + \log(2/\delta)} + H\sqrt{d\lambda}\right) \end{aligned}$$

D.4.6 Single-Step High Probability Upper Bound

Proof of Lemma 20

We have shown so far:

$$\begin{aligned} \phi(s,a)^\top\mathbf{w}_{t,h} - Q_{t,h}^\pi(s,a) \le\ & \texttt{bias}(t,h) + [\mathbb{P}_{t,h}(V_{t,h+1} - V_{t,h+1}^\pi)](s,a) \\ &+ \|\phi(s,a)\|_{\Sigma_{t,h}^{-1}\widetilde{\Sigma}_{t,h}\Sigma_{t,h}^{-1}}\left(CdH\sqrt{\log\left(dH\beta\frac{1}{\lambda(1-\eta)}\right) + \log(2/\delta)} + H\sqrt{d\lambda} + \sqrt{d\lambda}\right) \end{aligned}$$

so there exists an absolute constant $C' > 0$ such that

$$\begin{aligned} \phi(s,a)^\top\mathbf{w}_{t,h} - Q_{t,h}^\pi(s,a) \le\ & \texttt{bias}(t,h) + [\mathbb{P}_{t,h}(V_{t,h+1} - V_{t,h+1}^\pi)](s,a) \\ &+ \|\phi(s,a)\|_{\Sigma_{t,h}^{-1}\widetilde{\Sigma}_{t,h}\Sigma_{t,h}^{-1}}\left(C'dH\sqrt{\lambda}\sqrt{\log\left(dH\beta\frac{1}{\lambda(1-\eta)}\right) + \log(2/\delta)}\right) \end{aligned}$$

The missing ingredient to prove our key lemma is the choice of the parameter β

Now, we would like to find an appropriate choice of β such that

$$C'dH\sqrt{\lambda}\sqrt{\log\left(dH\beta\frac{1}{\lambda(1-\eta)}\right)+\log(2/\delta)} \leq \beta \tag{D.4.6}$$

First we set $\lambda = 1$. A good candidate for β is in the form of $\beta = c_\beta dH\sqrt{\iota}$ where $c_\beta > 1$ is an absolute constant and $\iota = \log\left(\frac{2dH}{\delta(1-\eta)}\right)$. With this choice, we obtain:

$$\begin{aligned}
C'dH\sqrt{\lambda}\sqrt{\log\left(dH\beta\frac{1}{\lambda(1-\eta)}\right)+\log(2/\delta)} &= C'dH\sqrt{\log(c_\beta dH\sqrt{\iota})+\iota} \\
&\leq C''dH\sqrt{\log(c_\beta)+\iota} \\
&\quad (C'' > 0 \text{ is an absolute constant}) \\
&\leq C''dH(\sqrt{\log(c_\beta)}+\sqrt{\iota})
\end{aligned}$$

Let $c_\beta > 1$ such that $C''(\sqrt{\log(c_\beta)}+\sqrt{\log(2)}) \leq \frac{c_\beta}{\sqrt{2}}\sqrt{\log(2)}$. In particular we have necessarily $\frac{c_\beta}{\sqrt{2}} \geq C''$ Therefore, we have:

$$\begin{aligned}
C''(\sqrt{\log(c_\beta)}+\sqrt{\iota}) &= C''(\sqrt{\log(c_\beta)}+\sqrt{\log(2)+(\iota-\log(2))}) \quad (\iota \geq \log(2)) \\
&\leq C''(\sqrt{\log(c_\beta)}+\sqrt{\log(2)}+\sqrt{(\iota-\log(2))}) \\
&\leq \frac{c_\beta}{\sqrt{2}}\sqrt{\log(2)}+C''\sqrt{(\iota-\log(2)} \\
&\leq \frac{c_\beta}{\sqrt{2}}(\sqrt{\log(2)}+\sqrt{(\iota-\log(2))} \\
&\leq \frac{c_\beta}{\sqrt{2}}\sqrt{2}\sqrt{\log(2)+\iota-\log(2)} \\
&\qquad (\,(a+b)^2 \leq 2(a^2+b^2) \Rightarrow a+b \leq \sqrt{2(a^2+b^2)}) \\
&\leq c_\beta\sqrt{\iota}
\end{aligned}$$

Therefore, with this choice of c_β and $\beta = c_\beta dH\sqrt{\iota}$, we obtain that

$$\left|\langle\phi(s,a),\mathbf{w}_{t,h}\rangle - Q^\pi_{t,h}(s,a) - [\mathbb{P}_{t,h}(V_{t,h+1}-V^\pi_{t,h+1})](s,a)\right| \leq \texttt{bias}(t,h) + \beta\left\|\phi(s,a)\right\|_{\Sigma^{-1}_{t,h}\widetilde{\Sigma}_{t,h}\Sigma^{-1}_{t,h}}.$$

D.4.7 Optimism

Proof of Lemma 21

We proceed by induction. By definition, we have $Q_{t,H+1} = Q^\star_{t,H+1} = 0$ and the desired statement trivially holds at step $H+1$. Now, assume that the

statement holds for $h+1$. Consider step h. By Lemma 20, we have

$$\left|\phi(s,a)^\top \mathbf{w}_{t,h} - Q^\star_{t,h}(s,a) - [\mathbb{P}_{t,h}(V_{t,h+1} - V^\star_{t,h+1})](s,a)\right| \le \mathbf{bias}(t,h) + \beta \left\|\phi(s,a)\right\|_{\Sigma^{-1}_{t,h}\widetilde{\Sigma}_{t,h}\Sigma^{-1}_{t,h}}$$

Moreover, we have

$$\begin{aligned} V^\star_{t,h+1}(s) - V_{t,h+1}(s) &= \max_{a\in\mathcal{A}} Q^\star_{t,h+1}(s,a) - \max_{a\in\mathcal{A}} Q_{t,h+1}(s,a) \\ &\le \max_{a\in\mathcal{A}} \left(Q^\star_{t,h+1}(s,a) - Q_{t,h+1}(s,a)\right) \\ &\le \sum_{h'=h+1}^{H} \mathbf{bias}(t,h) \qquad \text{(by the induction hypothesis)} \end{aligned}$$

Therefore, we obtain

$$\begin{aligned} Q^\star_{t,h}(s,a) &\le \phi(s,a)^\top \mathbf{w}_{t,h} + \beta \left\|\phi(s,a)\right\|_{\Sigma^{-1}_{t,h}\widetilde{\Sigma}_{t,h}\Sigma^{-1}_{t,h}} + [\mathbb{P}_{t,h}(V^\star_{t,h+1} - V_{t,h+1})](s,a) + \mathbf{bias}(t,h) \\ &\le \phi(s,a)^\top \mathbf{w}_{t,h} + \beta \left\|\phi(s,a)\right\|_{\Sigma^{-1}_{t,h}\widetilde{\Sigma}_{t,h}\Sigma^{-1}_{t,h}} + \sum_{h'=h}^{H} \mathbf{bias}(t,h) \end{aligned}$$

We have

$$\begin{aligned} Q^\star_{t,h}(s,a) &\le \phi(s,a)^\top \mathbf{w}_{t,h}\rangle + +\beta \left\|\phi(s,a)\right\|_{\Sigma^{-1}_{t,h}\widetilde{\Sigma}_{t,h}\Sigma^{-1}_{t,h}} + \sum_{h'=h}^{H} \mathbf{bias}(t,h) \\ &= Q_{t,h}(s,a) + \sum_{h'=h}^{H} \mathbf{bias}(t,h) \end{aligned}$$

D.5 Technical Lemmas

Lemma 23 (Concentration of weighted self-normalized processes [Russac et al., 2019]). *Let $\{\epsilon_t\}_{t=1}^\infty$ be a real-valued stochastic process with corresponding filtration $\{\mathcal{F}\}_{t=1}^\infty$. Let $\epsilon_t \mid \mathcal{F}_{t-1}$ be zero-mean and σ-subGaussiasn; i.e $\mathbb{E}[\epsilon_t \mid \mathcal{F}_{t-1}] = 0$ and*

$$\forall \lambda \in \mathbb{R}, \quad \mathbb{E}[e^{\lambda \epsilon_t} \mid \mathcal{F}_{t-1}] \le e^{\lambda^2\sigma^2/2}$$

Let $\{\phi_t\}_{t=1}^\infty$ be a predictable $\mathbb{R}^d$-valued stochastic process (i.e ϕ_t is $\mathcal{F}_{t-1}$-measurable) and $\{\omega_t\}_{t=0}^\infty$ be a sequence of predictable and positive weights. Let $\widetilde{\Sigma}_t = \sum_{s=1}^t \omega_s^2 \phi_s \phi_s^\top + \mu_t \cdot \mathbf{I}$ where $\{\mu_t\}_{t=1}^\infty$ a deterministic sequence of scalars. Then for any $\delta > 0$, with probability at least $1-\delta$, we have for all $t \ge 0$:

$$\left\|\sum_{s=1}^t \omega_s \phi_s \epsilon_s\right\|_{\widetilde{\Sigma}_t^{-1}} \le \sigma \sqrt{2\log\left(\frac{1}{\delta}\right) + \log\left(\frac{det(\widetilde{\Sigma}_t)}{\mu_t^d}\right)} \tag{D.5.1}$$

Lemma 24 (Determinant inequality for the weighted Gram matrix Russac et al. [2019]). *Let $\{\lambda_t\}_{t=0}^{\infty}$ and $\{\omega_t\}_{t=0}^{\infty}$ be a deterministic sequence of scalars. Let $\Sigma_t = \sum_{t=1}^{t} w_s \phi_s \phi_s^\top + \lambda_t \cdot \mathbf{I}$ be the weighted Gram matrix. Under the assumption $\forall t, \|\phi_t\| \leq 1$, the following holds*

$$det(\Sigma_t) \leq \left(\lambda_t + \frac{\sum_{s=1}^{t} \omega_s}{d}\right)^d \tag{D.5.2}$$

Lemma 25 (Covering Number of Euclidean Ball [Pollard, 1990]). *For any $\epsilon > 0$, the ϵ-covering number of the Euclidean ball in $\mathbb{R}^d$ with radius $R > 0$ is upper bounded by $(\frac{3R}{\epsilon})^d$*

Lemma 26 (Uniform concentration). *Let $\{s_t\}_{t=1}^{\infty}$ be a stochastic process on state state $\mathcal{S}$ with corresponding filtration $\{\mathcal{F}_t\}_{t=0}^{\infty}$. Let $\{\phi_t\}_{t=1}^{\infty}$ be an $\mathbb{R}^d$-valued stochastic process where ϕ_t is $\mathcal{F}_{t-1}$-measurable, and $\|\phi\| \leq 1$. Let $\widetilde{\Sigma}_t = \sum_{\tau=1}^{t} \eta^{-2\tau} \phi_\tau \phi_\tau^\top + \lambda \eta^{-2t} \cdot \mathbf{I}$. Then for any $\epsilon \in (0,1)$ and $\delta > 0$, with probability at least $1-\delta$, for all $t > 0$ and for all $\mathbf{w}, \mathbf{A} \in \mathcal{G}$ defined in (D.4.4), we have*

$$\begin{aligned} &\left\| \sum_{\tau=1}^{t} \eta^{-\tau} \phi_\tau \left(V^{\mathbf{w},\mathbf{A}}(s_\tau) - \mathbb{E}\left[V^{\mathbf{w},\mathbf{A}}(s_\tau) \mid \mathcal{F}_{\tau-1} \right] \right) \right\|_{\widetilde{\Sigma}_t^{-1}} \\ &\leq \mathcal{O}\left(dH \sqrt{\log\left(dH\beta_t \frac{1}{\lambda(1-\eta)} \right) + \log(1/\delta)} \right) \end{aligned} \tag{D.5.3}$$

where $V^{\mathbf{w},\mathbf{A}}$ is defined in Equation (D.4.3).

Proof. Let $t > 0$. For $\mathbf{w}, \mathbf{A} \in \mathcal{O}_t$ and $\tau \in [t]$ define

$$\epsilon_\tau^{\mathbf{w},\mathbf{A}} = V^{\mathbf{w},\mathbf{A}}(s_\tau) - \mathbb{E}\left[V^{\mathbf{w},\mathbf{A}}(s_\tau) \mid \mathcal{F}_{\tau-1} \right] \tag{D.5.4}$$

Then $\epsilon_\tau^{\mathbf{w},\mathbf{A}}$ defines a martingale difference sequence with filtration $\mathcal{F}_\tau$. Moreover, by the definition of $V^{\mathbf{w},\mathbf{A}}$, each $\epsilon_\tau^{\mathbf{w},\mathbf{A}}$ is bounded in absolute value by H, so that each $\epsilon_\tau^{\mathbf{w},\mathbf{A}}$ is H-subgaussian random variable.

So, by lemma 23, the $\epsilon_\tau^{\mathbf{w},\mathbf{A}}$ induce a self normalizing process so that for any $\delta > 0$, with probability at least $1-\delta$, we have for all $t > 0$:

$$\left\| \sum_{\tau=1}^{t} \eta^{-\tau} \phi_s \epsilon_\tau^{\mathbf{w},\mathbf{A}} \right\|_{\widetilde{\Sigma}_t^{-1}} \leq H \sqrt{2\log\left(\frac{1}{\delta}\right) + \log\left(\frac{\det(\widetilde{\Sigma}_t)}{(\lambda\eta^{-2t})^d} \right)} \tag{D.5.5}$$

$$\leq H \sqrt{2\log\left(\frac{1}{\delta}\right) + d\log\left(1 + \frac{1-\eta^{2t}}{\lambda d(1-\eta^2)}\right)} \tag{D.5.6}$$

The last step is due to $\det(\widetilde{\Sigma}_t) \leq \left(\lambda\eta^{-2t} + \frac{\eta^{-2t}-1}{d(1-\eta^{-d})}\right)^d$ by lemma 24.

Let $\mathcal{N}_\epsilon(\mathcal{G})$ be covering number of $\mathcal{G}$. So, by union bound, with probability δ. For all $\tilde{\mathbf{w}}, \tilde{\mathbf{A}}$ in the ϵ-covering of $\mathcal{G}$ that

$$\left\| \sum_{\tau=1}^{t} \eta^{-\tau} \phi_s \epsilon_\tau^{\tilde{\mathbf{w}},\tilde{\mathbf{A}}} \right\|_{\widetilde{\Sigma}_t^{-1}} \leq H \sqrt{2 \log\left(\frac{\mathcal{N}_\epsilon(\mathcal{G})}{\delta}\right) + d \log\left(1 + \frac{1-\eta^{2t}}{\lambda d(1-\eta^2)}\right)} \quad \text{(D.5.7)}$$

For any $(\mathbf{w}, \mathbf{A}) \in \mathcal{G}$, we choose a specific $(\tilde{\mathbf{w}}, \tilde{\mathbf{A}})$ in the ϵ-covering of $\mathcal{G}$ such that $\|\mathbf{w} - \tilde{\mathbf{w}}\| \leq \epsilon$ and $\left\|\mathbf{A} - \tilde{\mathbf{A}}\right\|_{\mathrm{F}} \leq \epsilon$.

$$\begin{aligned}
\left\| \sum_{\tau=1}^{t} \eta^{-\tau} \phi_s \epsilon_\tau^{\mathbf{w},\mathbf{A}} \right\|_{\widetilde{\Sigma}_t^{-1}} &\leq \left\| \sum_{\tau=1}^{t} \eta^{-\tau} \phi_s \epsilon_\tau^{\tilde{\mathbf{w}},\tilde{\mathbf{A}}} \right\|_{\widetilde{\Sigma}_t^{-1}} + \left\| \sum_{\tau=1}^{t} \eta^{-\tau} \phi_s \left(\epsilon_\tau^{\mathbf{w},\mathbf{A}} - \epsilon_\tau^{\tilde{\mathbf{w}},\tilde{\mathbf{A}}}\right) \right\|_{\widetilde{\Sigma}_t^{-1}} \\
&\leq H \sqrt{2 \log\left(\frac{\mathcal{N}_\epsilon(\mathcal{G})}{\delta}\right) + d \log\left(1 + \frac{1-\eta^{2t}}{\lambda d(1-\eta^2)}\right)} \\
&\quad + \left\| \sum_{\tau=1}^{t} \eta^{-\tau} \phi_s \left(\epsilon_\tau^{\mathbf{w},\mathbf{A}} - \epsilon_\tau^{\tilde{\mathbf{w}},\tilde{\mathbf{A}}}\right) \right\|_{\widetilde{\Sigma}_t^{-1}} \qquad \text{(D.5.8)}
\end{aligned}$$

We can bound

$$\begin{aligned}
\left\| \sum_{\tau=1}^{t} \eta^{-\tau} \phi_s \left(\epsilon_\tau^{\mathbf{w},\mathbf{A}} - \epsilon_\tau^{\tilde{\mathbf{w}},\tilde{\mathbf{A}}}\right) \right\|_{\widetilde{\Sigma}_t^{-1}} &\leq \frac{1}{\sqrt{\lambda}\eta^{-t}} \left\| \sum_{\tau=1}^{t} \eta^{-\tau} \phi_s \left(\epsilon_\tau^{\mathbf{w},\mathbf{A}} - \epsilon_\tau^{\tilde{\mathbf{w}},\tilde{\mathbf{A}}}\right) \right\| \\
&\leq \frac{1}{\sqrt{\lambda}\eta^{-t}} \cdot \frac{\eta^{-t}-1}{1-\eta} \sup_\tau |\epsilon_\tau^{\mathbf{w},\mathbf{A}} - \epsilon_\tau^{\tilde{\mathbf{w}},\tilde{\mathbf{A}}}| \\
&= \frac{1-\eta^t}{\sqrt{\lambda}(1-\eta)} \sup_\tau |\epsilon_\tau^{\mathbf{w},\mathbf{A}} - \epsilon_\tau^{\tilde{\mathbf{w}},\tilde{\mathbf{A}}}| \\
&\leq \frac{2(1-\eta^t)}{\sqrt{\lambda}(1-\eta)} \sup_\tau |V^{\mathbf{w},\mathbf{A}}(s_\tau) - V^{\tilde{\mathbf{w}},\tilde{\mathbf{A}}}(s_\tau)|
\end{aligned}$$

By the definition of $V^{\mathbf{w},\mathbf{A}}$, we have

$$\begin{aligned}
&\sup_{\tau} |V^{\mathbf{w},\mathbf{A}}(s_\tau) - V^{\tilde{\mathbf{w}},\tilde{\mathbf{A}}}(s_\tau)| \\
&\leq \sup_{s,a} \left| \left(\mathbf{w}^\top \phi(s,a) + \sqrt{\phi(s,a)^\top \mathbf{A} \phi(s,a)} \right) - \left(\tilde{\mathbf{w}}^\top \phi(s,a) + \sqrt{\phi(s,a)^\top \tilde{\mathbf{A}} \phi(s,a)} \right) \right| \\
&\leq \sup_{\phi \in \mathbb{R}^d : \|\phi\| \leq 1} \left| \left(\mathbf{w}^\top \phi + \sqrt{\phi^\top \mathbf{A} \phi} \right) - \left(\tilde{\mathbf{w}}^\top \phi + \sqrt{\phi^\top \tilde{\mathbf{A}} \phi} \right) \right| \\
&\leq \sup_{\phi \in \mathbb{R}^d : \|\phi\| \leq 1} |(\mathbf{w} - \tilde{\mathbf{w}})^\top \phi| + \sup_{\phi \in \mathbb{R}^d : \|\phi\| \leq 1} \sqrt{\phi^\top (\mathbf{A} - \tilde{\mathbf{A}}) \phi} \\
&= \|\mathbf{w} - \tilde{\mathbf{w}}\| + \sqrt{\left\|\mathbf{A} - \tilde{\mathbf{A}}\right\|} \\
&\leq \|\mathbf{w} - \tilde{\mathbf{w}}\| + \sqrt{\left\|\mathbf{A} - \tilde{\mathbf{A}}\right\|_{\mathrm{F}}} \leq \epsilon + \sqrt{\epsilon} \leq 2\sqrt{\epsilon}.
\end{aligned}$$

Therefore,

$$\left\| \sum_{\tau=1}^{t} \eta^{-\tau} \phi_s \left(\epsilon_\tau^{\mathbf{w},\mathbf{A}} - \epsilon_\tau^{\tilde{\mathbf{w}},\tilde{\mathbf{A}}} \right) \right\|_{\widetilde{\Sigma}_t^{-1}} \leq \frac{4(1-\eta^t)}{\sqrt{\lambda}(1-\eta)} \sqrt{\epsilon} \tag{D.5.9}$$

The ϵ-covering number of $\mathcal{G}$ as Euclidean ball in $\mathbb{R}^{d+d^2}$ of radius $2H\sqrt{\frac{d}{\lambda(1-\eta)}} + \frac{\beta^2\sqrt{d}}{\lambda}$ is bounded by Lemma 25 as $\left(3\left(2H\sqrt{\frac{d}{\lambda(1-\eta)}} + \frac{\beta^2\sqrt{d}}{\lambda}\right)/\epsilon\right)^{d+d^2}$. Now, combining Equations (D.5.8) and (D.5.9) we obtain:

$$\begin{aligned}
&\left\| \sum_{\tau=1}^{t} \eta^{-\tau} \phi_s \epsilon_\tau^{\mathbf{w},\mathbf{A}} \right\|_{\widetilde{\Sigma}_t^{-1}} \\
&\leq 2H\sqrt{2(d^2+d)\log\left(\frac{3}{\epsilon}\left(2H\sqrt{\frac{d}{\lambda(1-\eta)}} + \frac{\beta^2\sqrt{d}}{\lambda}\right)\right) + 2\log(\frac{1}{\delta}) + d\log\left(1 + \frac{1-\eta^{2t}}{\lambda d(1-\eta^2)}\right)} \\
&+ \frac{4\sqrt{\epsilon}(1-\eta^t)}{\sqrt{\lambda}(1-\eta)}
\end{aligned}$$

Finally by taking $\epsilon = \frac{\lambda(1-\eta)^2}{16}$ and keeping only dominant term for each parameter, we obtain.

$$\left\| \sum_{\tau=1}^{t} \eta^{-\tau} \phi_s \epsilon_\tau^{\mathbf{w},\mathbf{A}} \right\|_{\widetilde{\Sigma}_t^{-1}} \leq \mathcal{O}\left(dH\sqrt{\log\left(dH\beta \frac{1}{\lambda(1-\eta)} \right) + \log(1/\delta)} \right)$$

□

E Appendix for Chapter 7

E.1 PROOFS

The first proposition is a direct consequence of the definition (7.8.2) of successor states with a goal space G.

Proposition 12. *Let $\phi\colon S \times A \to G$ be a map to some goal space G.*

Let π be some policy, and let M^π be the successor state measure (7.8.2) of π in goal space G. Let m^π be the density of M^π with respect to some positive probability measure ρ on G.

Let $r\colon G \to \mathbb{R}$ be some function on G, and define the reward function $R(s,a) \triangleq r(\phi(s,a))$ on $S \times A$.

Then the Q-function Q^π of policy π for reward R is

$$\begin{aligned} Q^\pi(s,a) &= \int r(g)\, M^\pi(s,a,\mathrm{d}g) \\ &= \mathbb{E}_{g\sim\rho}\left[r(g)\, m^\pi(s,a,g)\right]. \end{aligned}$$

Proof of Proposition 12. For each time $t \geq 0$, let $P_t^\pi(s_0,a_0,\mathrm{d}g)$ be the probability distribution of $g = \phi(s_t,a_t)$ over trajectories of the policy π starting at (s_0,a_0) in the MDP. Thus, by the definition (7.8.2),

$$M^\pi(s,a,\mathrm{d}g) = \sum_{t\geq 0} \gamma^t P_t^\pi(s,a,\mathrm{d}g). \tag{E.1.1}$$

The Q-function of π for the reward R is by definition (the sums and integrals are finite since R is bounded)

$$\begin{aligned} Q^\pi(s,a) &= \sum_{t\geq 0} \gamma^t \mathbb{E}[R(s_t,a_t) \mid s_0 = s,\, a_0 = a,\, \pi] \\ &= \sum_{t\geq 0} \gamma^t \mathbb{E}[r(\phi(s_t,a_t)) \mid s_0 = s,\, a_0 = a,\, \pi] \\ &= \sum_{t\geq 0} \gamma^t \int_g r(g) P_t^\pi(s,a,\mathrm{d}g) \\ &= \int_g r(g) M^\pi(s,a,\mathrm{d}g) \\ &= \int_g r(g) m^\pi(s,a,g) \rho(\mathrm{d}g) \\ &= \mathbb{E}_{g\sim\rho}[r(g) m^\pi(s,a,g)] \end{aligned}$$

by definition of the density m^π. □

Proof of Theorems 1 and 2. Theorem 1 is a particular case of Theorem 2 ($\phi = \mathrm{Id}$ and $\bar{m} = 0$), so we only prove the latter.

Let $R(s,a) = r(\phi(s,a))$ be a reward function as in the theorem.

The Q-function of π for the reward R is, by Proposition 12,

$$Q^\pi(s,a) = \mathbb{E}_{g\sim\rho}[r(g)m^\pi(s,a,g)]. \tag{E.1.2}$$

The assumptions state that for any $z \in Z$, $m^{\pi_z}(s,a,g)$ is equal to $F(s,a,z)^\top B(g) + \bar{m}(s,z,g)$. Therefore, for any $z \in Z$ we have

$$\begin{aligned} Q^{\pi_z}(s,a) &= \mathbb{E}_{g\sim\rho}\left[r(g)F(s,a,z)^\top B(g) + r(g)\bar{m}(s,z,g)\right] \\ &= F(s,a,z)^\top \mathbb{E}_{g\sim\rho}\left[r(g)B(g)\right] + \mathbb{E}_{g\sim\rho}[r(g)\bar{m}(s,z,g)] \\ &= F(s,a,z)^\top z_R + \bar{V}^z(s) \end{aligned}$$

by definition of z_R and $\bar{V}$. This proves the claim (7.8.7) about Q-functions.

By definition, the policy π_z selects the action a that maximizes $F(s,a,z)^\top z$. Take $z = z_R$. Then

$$\pi_{z_R} = \operatorname*{argmax}_a F(s,a,z_R)^\top z_R \tag{E.1.3}$$

$$= \operatorname*{argmax}_a \left\{F(s,a,z_R)^\top z_R + \bar{V}^z(s)\right\} \tag{E.1.4}$$

since the last term does not depend on a.

This quantity is equal to $Q^{\pi_{z_R}}(s,a)$. Therefore,

$$\pi_{z_R} = \operatorname*{argmax}_a Q^{\pi_{z_R}}(s,a) \tag{E.1.5}$$

and by the above, $Q^{\pi_{z_R}}(s,a)$ is indeed equal to the Q-function of policy π_{z_R} for the reward R. Therefore, π_{z_R} and $Q^{\pi_{z_R}}$ constitute an optimal Bellman pair for reward R. Since $Q^{\pi_{z_R}}(s,a)$ is the Q-function of π_{Z_R}, it satisfies the Bellman equation

$$Q^{\pi_{z_R}}(s,a) = R(s,a) + \gamma \mathbb{E}_{s'|(s,a)} Q^{\pi_{z_R}}(s', \pi_{z_R}(s')) \tag{E.1.6}$$

$$= R(s,a) + \gamma \mathbb{E}_{s'|(s,a)} \max_{a'} Q^{\pi_{z_R}}(s',a') \tag{E.1.7}$$

by (E.1.5). This is the optimal Bellman equation for R, and π_{z_R} is the optimal policy for R.

We still have to prove the last statement of Theorem 2. Since π_{z_R} is an optimal policy for R, for any other policy π_z and state-action (s,a) we have

$$Q^{\pi_{z_R}}(s,a) \geq Q^{\pi_z}(s,a). \tag{E.1.8}$$

Using the formulas above for Q^π, with $\bar{m} = 0$, this rewrites as

$$F(s,a,z_R)^\top z_R \geq F(s,a,z)^\top z_R \tag{E.1.9}$$

as needed. Thus $F(s,a,z_R)^\top z_R \geq \sup_{z\in Z} F(s,a,z)^\top z_R$, and equality occurs by taking $z = z_R$. This ends the proof of Theorem 2. □

Proof of Proposition 6. Assume $d = \#S \times \#A$; extra dimensions can just be ignored by setting the extra components of F and B to 0.

With $d = \#S \times \#A$, we can index the components of Z by pairs (s,a).

First, let us set $B(s,a) \triangleq \mathbb{1}_{s,a}$.

Let $r\colon S \times A \to \mathbb{R}$ be any reward function. Let $z_R \in \mathbb{R}^{\#S\times\#A}$ be defined as in Theorem 1, namely, $z_R = \mathbb{E}_{(s,a)\sim\rho}[r(s,a)B(s,a)]$. With our choice of B, the components of z_R are $(z_R)_{s,a} = r(s,a)\rho(s,a)$. Since $\rho > 0$, the correspondence $r \leftrightarrow z_R$ is bijective.

Let us now define F. Take $z \in Z$. Since $r \leftrightarrow z_R$ is bijective, this z is equal to z_R for some reward function r. Let π_z be an optimal policy for this reward r in the MDP. Let M^{π_r} be the successor state measure of policy π_z, namely:

$$M^{\pi_z}(s,a,s',a') = \sum_{t\geq 0} \gamma^t \Pr\left((s_t,a_t) = (s',a') \mid (s_0,a_0) = (s,a),\ \pi_z\right). \tag{E.1.10}$$

Now define $F(s,a,z)$ by setting its (s',a') component to $M^{\pi_z}(s,a,s',a')/\rho(s',a')$ for each (s',a'):

$$F(s,a,z)_{s',a'} \triangleq M^{\pi_z}(s,a,s',a')/\rho(s',a'). \tag{E.1.11}$$

Then we have

$$\begin{aligned} F(s,a,z)^\top B(s',a') &= \sum_{s'',a''} F(s,a,z)_{s'',a''}\, B(s',a')_{s'',a''} \\ &= F(s,a,z)_{s',a'} = M^{\pi_z}(s,a,s',a')/\rho(s',a') \end{aligned} \tag{E.1.12}$$

because by our choice of B, $B(s',a')_{s''a''} = \mathbb{1}_{s'=s'',a'=a''}$.

Thus, $F(s,a,z)^\top B(s',a')$ is the density of the successor state measure M^{π_z} of policy π_z with respect to ρ, as needed.

We still have to check that π_z satisfies $\pi_z(s) = \operatorname{argmax} F(s,a,z)^\top z$ (since this is not how it was defined). Since π_z was defined as an optimal policy for the reward r associated with z, it satisfies

$$\pi_z(s) = \operatorname*{argmax}_a Q^{\pi_z}(s,a) \tag{E.1.13}$$

with $Q^{\pi_z}(s,a)$ the Q-function of policy π_z for the reward r. This Q-function is equal to the cumulated expected reward

$$\begin{aligned}
Q^{\pi_z}(s,a) &= \sum_{t\geq 0} \gamma^t \mathbb{E}\left[r(s_t,a_t) \mid s_0 = s,\, a_0 = a,\, \pi_z\right] \\
&= \sum_{t\geq 0} \gamma^t \sum_{s',a'} r(s',a') \Pr\left((s_t,a_t) = (s',a') \mid s_0 = s,\, a_0 = a,\, \pi_z\right) \\
&= \sum_{s',a'} r(s',a') \sum_{t\geq 0} \gamma^t \Pr\left((s_t,a_t) = (s',a') \mid s_0 = s,\, a_0 = a,\, \pi_z\right) \\
&= \sum_{s',a'} r(s',a') M^{\pi_z}(s,a,s',a') \\
&= \sum_{s',a'} r(s',a') F(s,a,z)_{s',a'}\, \rho(s',a') \\
&= F(s,a,z)^\top \left(\sum_{s'a'} r(s',a')\rho(s',a')\mathbb{1}_{s'a'}\right) \\
&= F(s,a,z)^\top z
\end{aligned}$$

since z is equal to $\mathbb{E}_{(s',a')\sim\rho}[r(s',a')B(s',a')]$. This proves that $\pi_z(s) = \operatorname{argmax}_a Q^{\pi_z(s,a)} = \operatorname{argmax}_a F(s,a,z)^\top z$. So this choice of F and B satisfies all the properties claimed. □

We will rely on the following two basic results in Q-learning.

Proposition 13 ($r \mapsto Q^\star$ is Lipschitz in sup-norm)**.** *Let r_1, $r_2 \colon S \times A \to \mathbb{R}$ be two bounded reward functions. Let $Q_1^\star$ and $Q_2^\star$ be the corresponding optimal Q-functions, and likewise for the V-functions. Then*

$$\sup_{S\times A} |Q_1^\star - Q_2^\star| \leq \frac{1}{1-\gamma} \sup_{S\times A} |r_1 - r_2| \text{ and } \sup_S |V_1^\star - V_2^\star| \leq \frac{1}{1-\gamma} \sup_{S\times A} |r_1 - r_2|\,. \tag{E.1.14}$$

Moreover for any policy π, we have

$$\sup_{S\times A} |Q_1^\pi - Q_2^\pi| \leq \frac{1}{1-\gamma} \sup_{S\times A} |r_1 - r_2| \text{ and } \sup_S |V_1^\pi - V_2^\pi| \leq \frac{1}{1-\gamma} \sup_{S\times A} |r_1 - r_2|\,. \tag{E.1.15}$$

Proof. Assume $\sup_{S\times A} |r_1 - r_2| \leq \epsilon$ for some $\epsilon \geq 0$.

For any policy π, let Q_1^π be its Q-function for reward r_1, and likewise for r_2. Let π_1 and π_2 be optimal policies for r_1 and r_2, respectively. Then for any

$(s,a) \in S \times A$,

$$\begin{aligned}
Q_1^\star(s,a) &= Q_1^{\pi_1}(s,a) \\
&\geq Q_1^{\pi_2}(s,a) \\
&= \sum_{t\geq 0} \gamma^t \mathbb{E}\left[r_1(s_t,a_t) \mid \pi_2,\, (s_0,a_0) = (s,a)\right] \\
&\geq \sum_{t\geq 0} \gamma^t \mathbb{E}\left[r_2(s_t,a_t) - \epsilon \mid \pi_2,\, (s_0,a_0) = (s,a)\right] \\
&= \sum_{t\geq 0} \gamma^t \mathbb{E}\left[r_2(s_t,a_t) \mid \pi_2,\, (s_0,a_0) = (s,a)\right] - \frac{\epsilon}{1-\gamma} \\
&= Q_2^\star(s,a) - \frac{\epsilon}{1-\gamma}
\end{aligned}$$

and likewise in the other direction, which ends the proof for Q-functions. The case of V-functions follows by restricting to the optimal actions at each state s.

Now, let π a policy. We have

$$\begin{aligned}
|Q_1^\pi(s,a) - Q_2^\pi(s,a)| = \Big| &\sum_{t\geq 0} \gamma^t \mathbb{E}\left[r_1(s_t,a_t) \mid \pi,\, (s_0,a_0) = (s,a)\right] \\
&- \sum_{t\geq 0} \gamma^t \mathbb{E}\left[r_2(s_t,a_t) \mid \pi,\, (s_0,a_0) = (s,a)\right] \Big| \\
&\leq \sum_{t\geq 0} \gamma^t \mathbb{E}\left[|r_2(s_t,a_t) - r_1(s_t,a_t)| \mid \pi,\, (s_0,a_0) = (s,a)\right] \\
&\leq \frac{1}{1-\gamma} \sup_{S\times A} |r_1 - r_2| \,.
\end{aligned}$$

The case of V-functions follows by taking the expectation over actions according to π. □

Proposition 14. *Let $f\colon S \times A \to \mathbb{R}$ be any function, and define a policy π_f by $\pi_f(s) \triangleq \operatorname{argmax}_a f(s,a)$. Let $r\colon S \times A \to \mathbb{R}$ be some bounded reward function. Let $Q^\star$ be its optimal Q-function, and let Q^{π_f} be the Q-function of π_f for reward r.*

Then

$$\sup_{S\times A} |f - Q^\star| \leq \frac{2}{1-\gamma} \sup_{S\times A} |f - Q^{\pi_f}| \tag{E.1.16}$$

and

$$\sup_{S\times A} |Q^{\pi_f} - Q^\star| \leq \frac{3}{1-\gamma} \sup_{S\times A} |f - Q^{\pi_f}| \tag{E.1.17}$$

Proof of Theorem 5. By construction of the Kantorovich–Rubinstein norm, the second claim of Theorem 5 is a particular case of the third claim, with $\|f\|_A \triangleq \max(\|f\|_\infty, \|f\|_{\mathrm{Lip}})$ and $\|\mu\|_B \triangleq \|\mu\|_{\mathrm{KR}}$.

Likewise, since m is the density of M with respect to ρ, the first claim is an instance of the third, by taking $\|f\|_A \triangleq \|f\|_\infty$ and $\|\mu\|_B \triangleq \left\|\frac{\mathrm{d}\mu}{\mathrm{d}\rho}\right\|_{L^1(\rho)}$. Therefore, we only prove the third claim.

Let $z \in Z$ and let $r\colon S \times A \to \mathbb{R}$ be any reward function. By Proposition 12 with $G = S \times A$ and $\phi = \mathrm{Id}$, the Q-function of policy Q^{π_z} for this reward is

$$Q^{\pi_z}(s,a) = \int r(s',a')\, M^{\pi_z}(s,a,\mathrm{d}s',\mathrm{d}a'). \tag{E.1.18}$$

Let $\varepsilon_z(s,a,\mathrm{d}s',\mathrm{d}a')$ be the difference of measures between the model $F^\top B\rho$ and M^{π_z}:

$$\varepsilon_z(s,a,\mathrm{d}s',\mathrm{d}a') \triangleq M^{\pi_z}(s,a,\mathrm{d}s',\mathrm{d}a') - \hat{M}^z(s,a,\mathrm{d}s',\mathrm{d}a') \tag{E.1.19}$$

$$= M^{\pi_z}(s,a,\mathrm{d}s',\mathrm{d}a') - F(s,a,z)^\top B(s',a')\rho(\mathrm{d}s',\mathrm{d}a'). \tag{E.1.20}$$

We want to control the optimality gap in terms of $\sup_{s,a} \|\varepsilon_z(s,a,\cdot)\|_B$.

By definition of ε_z,

$$Q^{\pi_z}(s,a) = \int r(s',a')F(s,a,z)^\top B(s',a')\rho(\mathrm{d}s',\mathrm{d}a') + \int r(s',a')\,\varepsilon_z(s,a,\mathrm{d}s',\mathrm{d}a') \tag{E.1.21}$$

$$= F(s,a,z)^\top z_R + \int r(s',a')\,\varepsilon_z(s,a,\mathrm{d}s',\mathrm{d}a') \tag{E.1.22}$$

since $z_R = \int r(s',a')B(s',a')\rho(\mathrm{d}s',\mathrm{d}a')$. Therefore,

$$\left|Q^{\pi_z}(s,a) - F(s,a,z)^\top z_R\right| = \left|\int r(s',a')\,\varepsilon_z(s,a,\mathrm{d}s',\mathrm{d}a')\right| \le \|r\|_A \,\|\varepsilon_z(s,a,\cdot)\|_B$$

for any reward r and any $z \in Z$ (not necessarily $z = z_R$).

Let $Q^\star$ be the optimal Q-function for reward r. Define $f(s,a) \triangleq F(s,a,z_R)^\top z_R$. By definition, the policy π_{z_R} is equal to $\mathrm{argmax}_a\, f(s,a)$. Therefore, by Proposition 14,

$$\sup_{S\times A} |Q^{\pi_{z_R}} - Q^\star| \le \frac{3}{1-\gamma} \sup_{S\times A} |f - Q^{\pi_{z_R}}|\,. \tag{E.1.23}$$

and

$$\sup_{S\times A} |f - Q^\star| \le \frac{2}{1-\gamma} \sup_{S\times A} |f - Q^{\pi_{z_R}}|\,. \tag{E.1.24}$$

But by the above,

$$\sup_{S\times A} |f - Q^{\pi_{z_R}}| = \sup_{S\times A} \left|F(s,a,z_R)^\top z_R - Q^{\pi_{z_R}}(s,a)\right| \le \|r\|_A \sup_{S\times A} \|\varepsilon_{z_R}(s,a,\cdot)\|_B\,.$$

Therefore, for any reward function r,

$$\sup_{S\times A} |Q^{\pi_{z_R}} - Q^\star| \le \frac{3\,\|r\|_A}{1-\gamma} \sup_{S\times A} \|\varepsilon_{z_R}(s,a,\cdot)\|_B\,. \tag{E.1.25}$$

This inequality transfers to the value functions, hence the result. In addition, using again $f(s,a) = F(s,a,z_R)^\top z_R$, we obtain

$$\sup_{S\times A} \left| F(s,a,z_R)^\top z_R - Q^\star(s,a) \right| \le \frac{2\,\|r\|_A}{1-\gamma} \sup_{S\times A} \|\varepsilon_{z_R}(s,a,\cdot)\|_B\,. \tag{E.1.26}$$

□

Proposition 15 (Pointwise optimality gap)**.** *Assume the state space is finite, and view rewards and Q-functions as vectors over state-actions.*

Let r_1 and r_2 be two reward functions, and let π_1 and π_2 be optimal policies for r_1 and r_2 respectively. Let P_1 and P_2 be the stochastic transition matrices over state-actions induced by π_1 and π_2.

Then the optimality gap of policy π_2 on reward r_1 is at most

$$0 \le Q_{r_1}^{\pi_1} - Q_{r_1}^{\pi_2} \le \left((\mathrm{Id} - \gamma P_1)^{-1} - (\mathrm{Id} - \gamma P_2)^{-1} \right) (r_1 - r_2) \tag{E.1.27}$$

where the equality holds componentwise viewing the Q-functions as vectors over state-actions.

Proof. This is a classical result. The inequality $0 \le Q_{r_1}^{\pi_1} - Q_{r_1}^{\pi_2}$ is trivial since π_1 is optimal for r_1 and the optimal policy is optimal at every state-action. Denote $M_1 \triangleq (\mathrm{Id} - \gamma P_1)^{-1}$ and likewise for M_2. Then for any reward function r one has $Q_r^{\pi_1} = M_1 r$ and likewise for M_2. Therefore

$$\begin{aligned} Q_{r_1}^{\pi_1} - Q_{r_1}^{\pi_2} &= M_1 r_1 - M_2 r_1 \\ &= M_1 r_1 - M_1 r_2 + M_1 r_2 - M_2 r_1 \\ &\le M_1 r_1 - M_1 r_2 + M_2 r_2 - M_2 r_1 \qquad \text{since } \pi_2 \text{ is optimal for } r_2 \\ &= (M_1 - M_2)(r_1 - r_2). \end{aligned}$$

□

Proof of Theorem 6. Let $f : S \times A \to \mathbb{R}$ be any function. Define the policy $\pi_f(s) \triangleq \mathrm{argmax}_a f(s,a)$. Define $r' \triangleq (\mathrm{Id} - \gamma P^{\pi_f}) f$. The equality $f = r' + \gamma P^{\pi_f} f$ can be rewritten as

$$f(s,a) = r'(s,a) + \gamma \mathbb{E}_{s' \sim P(\mathrm{d}s'|s,a)} f(s', \pi_f(s')). \tag{E.1.28}$$

But by definition of π_f, $\pi_f(s') = \mathrm{argmax}_{a'} f(s',a')$. Therefore

$$f(s,a) = r'(s,a) + \gamma \mathbb{E}_{s' \sim P(\mathrm{d}s'|s,a)} \max_{a'} f(s',a') \tag{E.1.29}$$

namely, f is the optimal Q-function for reward r', with π_f the corresponding optimal policy.

Let r be any reward function and let Q^{π_f} be the Q-function of π_f for r. By definition, it satisfies the Bellman equation $Q^{\pi_f} = r + \gamma P^{\pi_f} Q^{\pi_f}$, namely, $r = (\mathrm{Id} - \gamma P^{\pi_f}) Q^{\pi_f}$. Therefore,

$$r - r' = (\mathrm{Id} - \gamma P^{\pi_f})(Q^{\pi_f} - f). \tag{E.1.30}$$

For any policy π, denote $M^\pi \triangleq (\mathrm{Id} - \gamma P^\pi)^{-1}$. ($M^\pi$ is the successor measure seen as a matrix.) The Q-function of π for reward r is $M^\pi r$.

Since $\pi^\star$ is optimal for r, and π_f is optimal for r', Proposition 15 yields

$$\begin{aligned} 0 \leq Q^\star - Q^{\pi_f} &\leq (M^{\pi^\star} - M^{\pi_f})(r - r') \\ &= (M^{\pi^\star} - M^{\pi_f})(\mathrm{Id} - \gamma P^{\pi_f})(Q^{\pi_f} - f). \end{aligned}$$

Since M^{π_f} is the inverse of $\mathrm{Id} - \gamma P^{\pi_f}$ and likewise for $\pi^\star$, we have

$$\begin{aligned} (M^{\pi^\star} - M^{\pi_f})(\mathrm{Id} - \gamma P^{\pi_f}) &= M^{\pi^\star}(\mathrm{Id} - \gamma P^{\pi_f}) - \mathrm{Id} \\ &= M^{\pi^\star}(\mathrm{Id} - \gamma P^{\pi_f} - (\mathrm{Id} - \gamma P^{\pi^\star})) \\ &= \gamma M^{\pi^\star}(P^{\pi^\star} - P^{\pi_f}) \end{aligned}$$

and therefore

$$0 \leq Q^\star - Q^{\pi_f} \leq \gamma M^{\pi^\star}(P^{\pi^\star} - P^{\pi_f})(Q^{\pi_f} - f). \tag{E.1.31}$$

Now set

$$f(s,a) \triangleq F(s,a,z_R)^\top z_R \tag{E.1.32}$$

or in matrix notation, $f = F(z_R)^\top z_R$. Then $\pi_f = \pi_{z_R}$ by definition. Thus $Q^{\pi_f} = M^{\pi_{z_R}} r = m^{\pi_{z_R}} \operatorname{diag}(\rho) r$ in matrix notation. Moreover, $z_R = \mathbb{E}_\rho[B(s,a) r(s,a)] = B \operatorname{diag}(\rho) r$ in matrix notation. So $f = F(z_R)^\top B \operatorname{diag}(\rho) r$. Therefore,

$$Q^{\pi_f} - f = (m^{\pi_{z_R}} - F(z_R)^\top B) \operatorname{diag}(\rho) r \tag{E.1.33}$$

and

$$Q^\star - Q^{\pi_f} \leq \gamma M^{\pi^\star}(P_{\pi^\star} - P_{\pi_f}) \left(m^{\pi_{z_R}} - F(z_R)^\top B \right) \operatorname{diag}(\rho) r \tag{E.1.34}$$

and using $M^{\pi^\star} = \sum_{t \geq 0} \gamma^t P_{\pi_\star}^t$ provides the required inequality.

As a remark, the same proof works on continuous state spaces, by viewing all matrices as linear operators over functions on $S \times A$. □

Proof of Proposition 7. This is just a triangle inequality.

$$\begin{aligned}\|V_r^{\pi_{\hat{z}_R}} - V_r^\star\|_\infty &\leq \|V_r^{\pi_{\hat{z}_R}} - V_{\hat{r}}^{\pi_{\hat{z}_R}}\|_\infty + \|V_{\hat{r}}^{\pi_{\hat{z}_R}} - V_{\hat{r}}^\star\|_\infty + \|V_{\hat{r}}^\star - V_r^\star\|_\infty \\ &\leq \frac{\|r-\hat{r}\|_\infty}{1-\gamma} + \epsilon_{FB} + \frac{\|r-\hat{r}\|_\infty}{1-\gamma}.\end{aligned}$$

The last inequality follows from two facts: by assumption, the difference between the value function of $\pi_{\hat{z}_R}$ and the optimal value function $V_{\hat{r}}^\star$ is at most ε_{FB}, then by Proposition 13, the difference between $V_{\hat{r}}^\star$ and $V_r^\star$ as well as the difference between $V_r^{\pi_{\hat{z}_R}}$ and $V_{\hat{r}}^{\pi_{\hat{z}_R}}$ are bounded by $\frac{1}{1-\gamma}\sup_{S\times A}|\hat{r}-r|$. □

Proof of Theorem 7. We proceed by building a reward function $\hat{r}$ corresponding to $\hat{z}_R$. Then we will bound $\hat{r}-r$ and apply Proposition 7.

First, by Remark 4, up to reducing d, we can assume that B is $L^2(\rho)$-orthonormal.

For any function $\phi\colon (s,a)\to\mathbb{R}$, define $z_\phi \triangleq \mathbb{E}_{(s,a)\sim\rho}[\phi(s,a)B(s,a)]$. For each $z\in Z$, define ϕ_z via $\phi_z(s,a) \triangleq B(s,a)^\top z$. Then, if B is $L^2(\rho)$-orthonormal, we have $z_{\phi_z} = z$. (Indeed, $z_{\phi_z} = \mathbb{E}[(B(s,a)^\top z)B(s,a)] = \mathbb{E}[B(s,a)(B(s,a)^\top z)] = \left(\mathbb{E}[B(s,a)B(s,a)^\top]\right)z$.)

Define the function

$$\hat{r} \triangleq r + \phi_{\hat{z}_R - z_R} \tag{E.1.35}$$

using the functions ϕ_z defined above. By construction, $z_{\hat{r}} = z_R + z_{\phi_{\hat{z}_R - z_R}} = \hat{z}_R$. Therefore, the policy $\pi_{\hat{z}_R}$ associated to $\hat{z}_R$ is the policy associated to the reward $\hat{r}$.

We will now apply Proposition 7 to r and $\hat{r}$. For this, we need to bound $\|\phi_{\hat{z}_R - z_R}\|_\infty$.

Let $B_1, B_2, \ldots, B_d$ be the components of B as functions on $S\times A$. For any $z\in Z$, we have

$$\|\phi_z\|_{L^2(\rho)}^2 = \left\|\sum_i z_i B_i\right\|_{L^2(\rho)}^2 = \sum_i z_i^2 = \|z\|^2 \tag{E.1.36}$$

since the B_i are $L^2(\rho)$-orthonormal. Moreover, by construction, ϕ_z lies in the linear span $\langle B\rangle$ of the functions (B_i). Therefore

$$\|\phi_z\|_\infty \leq \zeta(B)\,\|\phi_z\|_{L^2(\rho)} = \zeta(B)\,\|z\| \tag{E.1.37}$$

by the definition of $\zeta(B)$ (Definition 6).

Therefore,

$$\|\phi_{\hat{z}_R - z_R}\|_\infty \leq \zeta(B)\,\|\hat{z}_R - z_R\|\,. \tag{E.1.38}$$

Let us now bound $\hat{z}_R - z_R$:

$$\begin{aligned}
&\mathbb{E}\left[\|\hat{z}_R - z_R\|^2\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\|\hat{z}_R - z_R\|^2 \mid (s_i, a_i)\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\|\hat{z}_R - \mathbb{E}[\hat{z}_R \mid (s_i, a_i)]\|^2 + \|\mathbb{E}[\hat{z}_R \mid (s_i, a_i)] - z_R\|^2 \mid (s_i, a_i)\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\left\|\frac{1}{N}\sum_i (\hat{r}_i - r(s_i, a_i))B(s_i, a_i)\right\|^2 + \left\|\frac{1}{N}\sum_i r(s_i, a_i)B(s_i, a_i) - z_R\right\|^2 \mid (s_i, a_i)\right]\right]
\end{aligned}$$

The first term satisfies

$$\begin{aligned}
\mathbb{E}\left[\left\|\frac{1}{N}\sum_i (\hat{r}_i - r(s_i, a_i))B(s_i, a_i)\right\|^2 \mid (s_i, a_i)\right] &= \frac{1}{N^2}\sum_i \mathbb{E}\left[(\hat{r}_i - r(s_i, a_i))^2 \|B(s_i, a_i)\|^2\right] \\
&\leq \frac{1}{N^2}\sum_i v \|B(s_i, a_i)\|^2
\end{aligned}$$

because the $\hat{r}_i$ are independent conditionally to (s_i, a_i), and because B is deterministic. The expectation of this over (s_i, a_i) is

$$\mathbb{E}\left[\frac{1}{N^2}\sum_i v \|B(s_i, a_i)\|^2\right] = \frac{v}{N}\mathbb{E}_{(s,a)\sim\rho}\|B(s, a)\|^2 = \frac{v}{N}\|B\|^2_{L^2(\rho)} \tag{E.1.39}$$

which is thus a bound on the first term.

The second term satisfies

$$\mathbb{E}\left[\left\|\frac{1}{N}\sum_i r(s_i, a_i)B(s_i, a_i) - z_R\right\|^2\right] = \frac{1}{N}\left\|r(s, a)B(s, a) - \mathbb{E}_\rho[r(s, a)B(s, a)]\right\|^2_{L^2(\rho)} \tag{E.1.40}$$

since the (s_i, a_i) are independent with distribution ρ. By the Cauchy–Schwarz inequality (applied to each component of B), this is at most

$$\frac{1}{N}\left\|r(s, a) - \mathbb{E}_\rho r\right\|^2_{L^2(\rho)}\|B\|^2_{L^2(\rho)}. \tag{E.1.41}$$

Therefore,

$$\mathbb{E}\left[\|\hat{z}_R - z_R\|^2\right] \leq \left(v + \left\|r(s, a) - \mathbb{E}_\rho r\right\|^2_{L^2(\rho)}\right)\frac{\|B\|^2_{L^2(\rho)}}{N}. \tag{E.1.42}$$

Since B is orthonormal in $L^2(\rho)$, we have $\|B\|^2_{L^2(\rho)} = d$. Putting everything together, we find

$$\mathbb{E}\left[\|\hat{r} - r\|^2_\infty\right] \leq \frac{\zeta(B)\, d}{N}\left(v + \left\|r(s, a) - \mathbb{E}_\rho r\right\|^2_{L^2(\rho)}\right). \tag{E.1.43}$$

Therefore, by the Markov inequality, for any $\delta > 0$, with probability $1 - \delta$,

$$\|\hat{r} - r\|_\infty \leq \sqrt{\frac{\zeta(B)\, d}{N\delta} \left(v + \left\| r(s,a) - \mathbb{E}_\rho r \right\|^2_{L^2(\rho)} \right)} \tag{E.1.44}$$

hence the conclusion by Proposition 7. □

Proof of Theorem 8. Let

$$m(s,a,s',a') \triangleq \sum_{t \geq 0} \gamma^t \frac{P_t(\mathrm{d}s', \mathrm{d}a' | s, a, \pi_z)}{\rho(\mathrm{d}s', \mathrm{d}a')} \tag{E.1.45}$$

so that

$$\ell(F,B) = \int \left| F(s,a,z)^\top B(s',a') - m(s,a,s',a') \right|^2 \rho(\mathrm{d}s, \mathrm{d}a) \rho(\mathrm{d}s', \mathrm{d}a'). \tag{E.1.46}$$

Let us first take the derivative with respect to the parameters of F. This is 0 by assumption, so we find

$$0 = \int \partial_\theta F(s,a,z)^\top B(s',a') \left(F(s,a,z)^\top B(s',a') - m(s,a,s',a') \right) \rho(\mathrm{d}s, \mathrm{d}a) \rho(\mathrm{d}s', \mathrm{d}a') \tag{E.1.47}$$

$$= \int \partial_\theta F(s,a,z)^\top G(s,a) \rho(\mathrm{d}s, \mathrm{d}a) \tag{E.1.48}$$

where

$$G(s,a) \triangleq \int B(s',a') \left(F(s,a,z)^\top B(s',a') - m(s,a,s',a') \right) \rho(\mathrm{d}s', \mathrm{d}a') \tag{E.1.49}$$

Since the model is overparameterized, we can realize any L^2 function $f(s,a)$ as the derivative $\partial_\theta F(s,a,z)$ for some direction θ. Therefore, the equation $0 = \int \partial_\theta F(s,a,z)^\top G(s,a) \rho(\mathrm{d}s, \mathrm{d}a)$ implies that $G(s,a)$ is $L^2(\rho)$-orthogonal to any function $f(s,a)$ in $L^2(\rho)$. Therefore, $G(s,a)$ vanishes ρ-almost everywhere, namely

$$\int B(s',a') F(s,a,z)^\top B(s',a') \rho(\mathrm{d}s', \mathrm{d}a') = \int B(s',a') m(s,a,s',a') \rho(\mathrm{d}s', \mathrm{d}a') \tag{E.1.50}$$

Now, since $F(s,a,z)^\top B(s',a')$ is a real number, $F(s,a,z)^\top B(s',a') = B(s',a')^\top F(s,a,z)$. Therefore, the right-hand-side above rewrites as

$$\int B(s',a') B(s',a')^\top F(s,a,z) \rho(\mathrm{d}s', \mathrm{d}a') = (\operatorname{Cov} B) F(s,a,z) \tag{E.1.51}$$

so that

$$(\operatorname{Cov} B) F(s,a,z) = \int B(s',a') m(s,a,s',a') \rho(\mathrm{d}s', \mathrm{d}a'). \tag{E.1.52}$$

Unfolding the definition of m yields the statement for F. The proof for B is similar. □

Proof of Theorem 9. According to the proof of Theorem 8, if F achieves a local extremum of $\ell(F, B)$ given a fixed B, we have

$$(\operatorname{Cov} B)F(s,a,z) = \int B(s',a')m^{\pi_z}(s,a,s',a')\rho(\mathrm{d}s',\mathrm{d}a') \tag{E.1.53}$$

where $m^{\pi_z}(s,a,s',a') = \sum_{t\geq 0}\gamma^t \frac{P_t(\mathrm{d}s',\mathrm{d}a'|s,a,\pi_z)}{\rho(\mathrm{d}s',\mathrm{d}a')}$ is the successor state density induced by the policy π_z.

Let $r\colon S\times A \to \mathbb{R}$ a bounded reward function that lies in the span of B i.e there exists $\omega \in \mathbb{R}^d$ such that $r(s,a) = B(s,a)^\top\omega$ for any state-action pair (s,a). This implies that $(\operatorname{Cov} B)\omega = \mathbb{E}_{(s,a)\sim\rho}[B(s,a)B(s,a)^\top\omega] = \mathbb{E}_{(s,a)\sim\rho}[B(s,a)r(s,a)] = z_R$, by definition of z_R.

Therefore,

$$\begin{aligned}
F(s,a,z_R)^\top z_R &= F(s,a,z_R)^\top(\operatorname{Cov} B)\omega \\
&= ((\operatorname{Cov} B)F(s,a,z_R))^\top\omega \\
&= \left(\int B(s',a')^\top m^{\pi_{z_R}}(s,a,s',a')\rho(\mathrm{d}s',\mathrm{d}a')\right)\omega \\
&= \int B(s',a')^\top\omega\, m^{\pi_{z_R}}(s,a,s',a')\rho(\mathrm{d}s',\mathrm{d}a') \\
&= \int r(s',a')m^{\pi_{z_R}}(s,a,s',a')\rho(\mathrm{d}s',\mathrm{d}a') \\
&= Q^{\pi_{z_R}}(s,a)
\end{aligned}$$

Therefore, π_{z_R} is the greedy policy with respect to its own Q-value. We conclude that π_{z_R} is optimal for r.

Now, let $r\colon S\times A\to\mathbb{R}$ be an arbitrary bounded reward function, and let r_B be the $L^2(\rho)$-projection of r onto the span of B. Both r and r_B share the same $z_R = \mathbb{E}_{(s,a)\sim\rho}[r(s,a)B(s,a)] = \mathbb{E}_{(s,a)\sim\rho}[r_B(s,a)B(s,a)]$. According to the first part of our proof, π_{z_R} is optimal for r_B since r_B lies in the span of B. Denote by the subscript r in V_r the reward function that a value function corresponds to. By Proposition 15,

$$0 \leq V_r^\star - V_r^{\pi_{z_R}} \leq (\mathrm{Id}-\gamma P^{\pi^\star})^{-1}(r-r_B) - (\mathrm{Id}-\gamma P^{\pi_{z_R}})^{-1}(r-r_B) \tag{E.1.54}$$

hence, taking norms,

$$\|V^{\pi_{z_R}} - V^\star\|_\infty \leq \left\|(\mathrm{Id}-\gamma P^{\pi^\star})^{-1}(r-r_B)\right\|_\infty + \left\|(\mathrm{Id}-\gamma P^{\pi_{z_R}})^{-1}(r-r_B)\right\|_\infty \tag{E.1.55}$$

as needed.

The bound with $\frac{2}{1-\gamma}$ follows by noting that $(\mathrm{Id}-\gamma P^\pi)^{-1}$ is bounded by $\frac{1}{1-\gamma}$ in L^∞ norm for any policy π. □

Proof of Proposition 8. From the definition (7.8.33) of $\hat{z}_R$ and the definition (7.8.34) of $\hat{r}$, we find

$$\begin{aligned}\hat{z}_R &= \mathbb{E}_\rho[B(s,a)\hat{r}(s,a)] \\ &= \mathbb{E}_\rho[B(s,a)B(s,a)^\top w] \\ &= (\mathrm{Cov}_\rho B)w\end{aligned}$$

hence the result given the expression (7.8.34) for w.

If $\rho = \rho_{\text{test}}$, then the covariances cancel out: we find $\hat{z}_R = (\mathrm{Cov}_\rho B)w = (\mathrm{Cov}_\rho B)(\mathrm{Cov}_{\rho_{\text{test}}} B)^{-1}\mathbb{E}_{\rho_{\text{test}}}[r(s,a)B(s,a)] = \mathbb{E}_{\rho_{\text{test}}}[r(s,a)B(s,a)] = \mathbb{E}_\rho[r(s,a)B(s,a)] = z_R$.

If r is linear in B, then the linear regression model does not depend on the data distribution ρ_{test} used: if $r(s,a) = B(s,a)^\top w_0$ then $w = w_0$ for any ρ_{test}, as long as $\mathrm{Cov}_{\rho_{\text{test}}} B$ is invertible. In that case, both z_R and $\hat{z}_R$ are equal to $(\mathrm{Cov}_\rho B)w_0$. □

E.2 Experimental Setup

In this section we provide additional information about our experiments.

E.2.1 Environments

- **Discrete maze:** is the 11×11 classical tabular gridworld with foor rooms. States are represented by one-hot unit vectors, $S = \{0,1\}^{121}$. There are five available actions , $A = \{\texttt{left}, \texttt{right}, \texttt{up}, \texttt{down}, \texttt{do nothing}\}$. The dynamics are deterministic and the walls are impassable.

- **Continuous maze:** is a two dimensional environment with impassable walls. States are represented by their Cartesian coordinates $(x,y) \in S = [0,1]^2$. There are five available actions, $A = \{\texttt{left}, \texttt{right}, \texttt{up}, \texttt{down}, \texttt{do nothing}\}$. The execution of one of the actions moves the agent 0.1 units in the desired direction, and normal random noise with zero mean and standard deviation 0.01 is added to the position of the agent (that is, a move along the x axis would be $x' = x \pm 0.1 + \mathcal{N}(0, 0.01)$, where $\mathcal{N}(0, 0.01)$ is a normal variable with mean 0 and standard deviation 0.01). If after a move the agent ends up outside of $[0,1]^2$, the agent's position is clipped (e.g if $x < 0$ then we set $x = 0$). If a move make the agent cross an interior wall, this move is undone. For all algorithms, we convert a state $s = (x,y)$ into feature vector $\phi(s) \in \mathbb{R}^{441}$ by computing the activations of a regular 21×21 grid of radial basis functions at the point (s,y). Especially, we use Gaussian functions: $\phi(s) =$

$\left(\exp(-\frac{(x-x_i)^2+(y-y_i)^2}{\sigma}), \ldots, \exp(-\frac{(x-x_{441})^2+(y-y_{441})^2}{2\sigma^2})\right)$ where (x_i, y_i) is the center of the i^{th} Gaussian and $\sigma = 0.05$.

- **FeatchReach:** is a variant of the simulated robotic arm environment from Plappert et al. [2018] using discrete actions instead of continuous actions. States are 10-dimensional vectors consisting of positions and velocities of robot joints. We discretise the original 3-dimensional action space into 6 possible actions using action stepsize of 1 (The same way as in https://github.com/paulorauber/hpg, the implementation of hindsight policy gradient Rauber et al. [2018]). The goal space is 3-dimensional space representing of the position of the object to reach.

- **Ms. Pacman:** is a variant of the Atari 2600 game Ms. Pacman Bellemare et al. [2013b], where an episode ends when the agent is captured by a monster Rauber et al. [2018]. States are obtained by processing the raw visual input directly from the screen. Frames are preprocessed by cropping, conversion to grayscale and downsampling to 84×84 pixels. A state s_t is the concatenation of $(x_{t-12}, x_{t-8}, x_{t-4}, x_t)$ frames, i.e. an $84 \times 84 \times 4$ tensor. An action repeat of 12 is used. As Ms. Pacman is not originally a multi-goal domain, we define the set of goals as the set of the 148 reachable coordinate pairs (x, y) on the screen; these can be reached only by learning to avoid monsters. In contrast with Rauber et al. [2018], who use a heuristic to find the agent's position from the screen's pixels, we use the Atari annotated RAM interface wrapper Anand et al. [2019].

E.2.2 Architectures

We use the same architecture for discrete maze, continuous maze and FeatchReach. Both forward and backward networks are represented by a feedforward neural network with three hidden layers, each with 256 ReLU units. The forward network receives a concatenation of a state and a z vector as input and has $|A| \times d$ as output dimension. The backward network receives a state as input (or gripper's position for FeatchReach) and has d as output dimension. For goal-oriented DQN, the Q-value network is also a feedforward neural network with three hidden layers, each with 256 ReLU units. It receives a concatenation of a state and a goal as input and has $|A|$ as output dimension.

For Ms. Pacman, the forward network is represented by a convolutional neural network given by a convolutional layer with 32 filters (8×8, stride 4); convolutional layer with 64 filters (4×4, stride 2); convolutional layer with 64 filters (3×3, stride 1); and three fully-connected layers, each with 256 units. We use ReLU as activation function. The z vector is concatenated with the output of the third convolutional layer. The output dimension of the final linear layer is

$|A| \times d$. The backward network acts only on agent's position, a 2-dimensional input. It is represented by a feedforward neural network with three hidden layers, each with 256 ReLU units. The output dimension is d. For goal-oriented DQN, the Q-value network is represented by a convolutional neural network with the same architecture as the one of the forward network. The goal's position is concatenated with the output of the third convolutional layer. The output dimension of the final linear layer is $|A|$.

E.2.3 Implementation Details

For all environments, we run the algorithms for 800 epochs. Each epoch consists of 25 cycles where we interleave between gathering some amount of transitions, to add to the replay buffer $\mathcal{D}$ (old transitions are thrown when we reach the maximum of its size), and performing 40 steps of stochastic gradient descent on the model parameters. To collect transitions, we generate episodes using some behavior policy. For both mazes, we use a uniform policy while for FetchReach and Ms. Pacman, we use an ϵ-greedy policy ($\varepsilon = 0.2$) with respect to the current approximation $F(s, a, z)^\top z$ for a sampled z. At evaluation time, ε-greedy policies are also used, with a smaller $\varepsilon = 0.02$ for all environments except from discrete maze where we use Boltzmann policy with temperature $\tau = 1$. We train each models for three different random seeds.

For generality, we will keep using the notation $B(s, a)$ while in our experiments B acts only on $\phi(s, a)$, a part of the state-action space. For discrete and continuous mazes, $\phi(s, a) = s$, for FetchReach, $\phi(s, a)$ the position of arm's gripper and for Ms. Pacman, $\phi(s, a)$ is the 2-dimensional position (x, y) of the agent on the screen.

We denote by θ and ω the parameters of forward and backward networks respectively and θ^- and ω^- the parameters of their corresponding target networks. Both θ^- and ω^- are updated after each cycle using Polyak averaging; i.e $\theta^- \leftarrow \alpha\theta^- + (1 - \alpha)\theta$ and $\omega^- \leftarrow \alpha\omega^- + (1 - \alpha)\omega$ where $\alpha = 0.95$ is the Polyak coefficient.

During training, we sample z from a rescaled Gaussian that we denote ν. Especially, we sample a d-dimensional standard Gaussian variable $x \sim \mathcal{N}(0, \mathrm{Id}) \in \mathbb{R}^d$ and a scalar centered Cauchy variable $u \in \mathbb{R}$ of scale 0.5, then we set $z = \sqrt{d} u \frac{x}{\|x\|}$. We use a Cauchy distribution to ensure that the norm of z spans the non-negative real numbers space while having a heavy tail: with a pure Gaussian, the norm of z would be very concentrated around a single value. We also scale by $\sqrt{d}$ to ensure that each component of z has an order of magnitude of 1.

Before being fed to F, z is preprocessed by $z \leftarrow \frac{z}{\sqrt{1+\|z\|_2^2/d}}$; this way, z ranges over a bounded set in $\mathbb{R}^d$, and this takes advantage of optimal policies being equal for a reward R and for λR with $\lambda > 0$.

To update network parameters, we compute an empirical loss by sampling 3 mini-batches, each of size $b = 128$, of transitions $\{(s_i, a_i, s_{i+1})\}_{i \in I} \subset \mathcal{D}$, of target state-action pairs $\{(s'_i, a'_i)\}_{i \in I} \subset \mathcal{D}$ and of $\{z_i\}_{i \in I} \sim \nu$:

$$\mathscr{L}(\theta, \omega) = -\frac{1}{b} \sum_{i \in I} F_\theta(s_i, a_i, z_i)^\top B_\omega(s_i, a_i) \tag{E.2.1}$$

$$\frac{1}{2b^2} \sum_{i,j \in I^2} \left(F_\theta(s_i, a_i, z_i)^\top B_\omega(s'_j, a'_j) - \gamma \sum_{a \in A} \pi_{z_i}(a \mid s_{i+1}) \cdot F_{\theta^-}(s_{i+1}, a, z_i)^\top B_{\omega^-}(s'_j, a'_j) \right)^2$$

where we use the Boltzmann policy $\pi_{z_i}(\cdot \mid s_{i+1}) = \texttt{softmax}(F_{\theta^-}(s_{i+1}, \cdot, z_i)^\top z_i / \tau)$ with fixed temperature $\tau = 200$ to avoid the instability and discontinuity caused by the argmax operator.

Since there is unidentifiability between F and B (Appendix, Remark 4), we include a gradient to make B closer to orthonormal, $\mathbb{E}_{(s,a) \sim \rho} B(s,a) B(s,a)^\top \approx \mathrm{Id}$:

$$\frac{1}{4} \partial_\omega \left\| \mathbb{E}_{(s,a) \sim \rho} B_\omega(s,a) B_\omega(s,a)^\top - \mathrm{Id} \right\|^2 =$$
$$\mathbb{E}_{(s,a) \sim \rho, (s',a') \sim \rho} \partial_\omega B_\omega(s,a)^\top \left(B_\omega(s,a)^\top B_\omega(s',a') \cdot B_\omega(s',a') - B(s,a) \right) \tag{E.2.2}$$

To compute an unbiased estimate of the latter gradient, we use the following auxiliary empirical loss:

$$\mathscr{L}_{\texttt{reg}}(\omega) = -\frac{1}{b} \sum_{i \in I} B_\omega(s_i, a_i)^\top \texttt{stop-gradient}(B_\omega(s_i, a_i)) \tag{E.2.3}$$

$$\frac{1}{b^2} \sum_{i,j \in I^2} B_\omega(s_i, a_i)^\top \texttt{stop-gradient}(B_\omega(s'_j, a'_j)) \cdot \texttt{stop-gradient}(B_\omega(s_i, a_i)^\top B_\omega(s'_j, a'_j))$$

Finally, we use the Adam optimizer and we update θ and ω by taking a gradient step on $\mathscr{L}(\theta, \omega)$ and $\mathscr{L}(\theta, \omega) + \lambda \cdot \mathscr{L}_{\texttt{reg}}(\omega)$ respectively, where λ is a regularization coefficient that we set to 1 for all experiments.

We summarize the hyperparameters used for FB algorithm and goal-oriented DQN in table E.1 and E.2 respectively.

E.2.4 Experimental results

In this section, we provide additional experimental results.

Goal-Oriented Setup: Quantitative Comparisons

More Complex Rewards: Qualitative Results

Embedding Visualization

Hyperparameters	Discrete Maze	Continuous Maze	FetchReach	Ms. Pacman
number of cycles per epoch	25	25	25	25
number of episodes per cycles	4	4	2	2
number of timesteps per episode	50	30	50	50
number of updates per cycle	40	40	40	40
exploration ε	1	1	0.2	0.2
evaluation ε	Boltzman with $\tau = 1$	0.02	0.02	0.02
temperature τ	200	200	200	200
learning rate	0.001	0.0005	0.0005	0.0001 if $d = 100$ else 0.0005
mini-batch size	128	128	128	128
regularization coefficient λ	1	1	1	1
Polyak coefficient α	0.95	0.95	0.95	0.95
discount factor γ	0.99	0.99	0.9	0.9
replay buffer size	10^6	10^6	10^6	10^6

Table E.1: Hyperparameters of the FB algorithm

Hyperparameters	Discrete Maze	Continuous Maze	FetchReach	Ms. Pacman
number of cycles per epoch	25	25	25	25
number of episodes per cycles	4	4	2	2
number of timesteps per episode	50	30	50	50
number of updates per cycle	40	40	40	40
exploration ε	0.2	0.2	0.2	0.2
evaluation ε	Boltzman with $\tau = 1$	0.02	0.02	0.02
learning rate	0.001	0.0005	0.0005	0.0005
mini-batch size	128	128	128	128
Polyak coefficient α	0.95	0.95	0.95	0.95
discount factor γ	0.99	0.99	0.9	0.9
replay buffer size	10^6	10^6	10^6	10^6
ratio of hindsight replay	-	-	-	0.8

Table E.2: Hyperparameters of the goal-oriented DQN algorithm

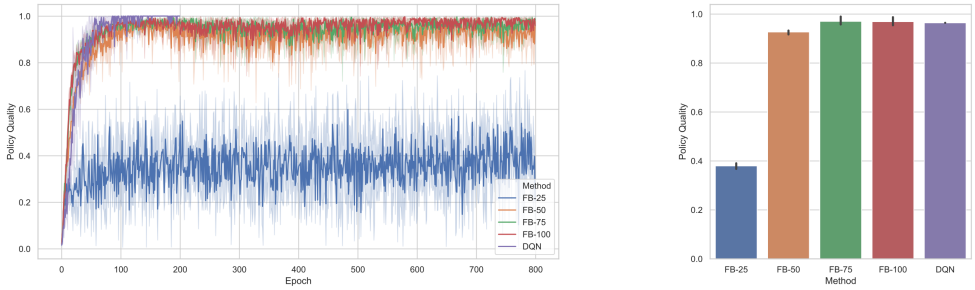

Figure E.1: Discrete maze: Comparative performance of FB for different dimensions and DQN. **Left**: the policy quality averaged over 20 randomly selected goals as function of the training epochs. **Right**: the policy quality averaged over the goal space after 800 training epochs.

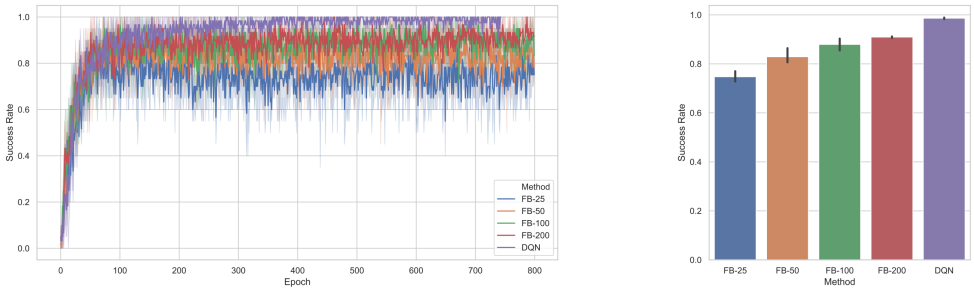

Figure E.2: Continuous maze: Comparative performance of FB for different dimensions and DQN. **Left**: the success rate averaged over 20 randomly selected goals as function of the training epochs. **Right**: the success rate averaged over 1000 randomly sampled goals after 800 training epochs.

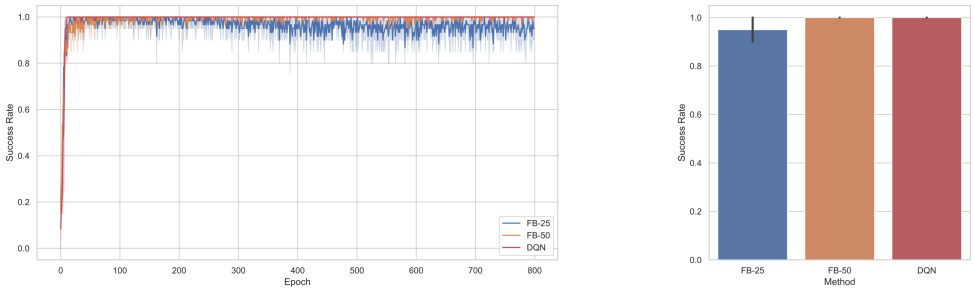

Figure E.3: FetchReach: Comparative performance of FB for different dimensions and DQN. **Left**: the success rate averaged over 20 randomly selected goals as function of the training epochs. **Right**: the success rate averaged over 1000 randomly sampled goals after 800 training epochs.

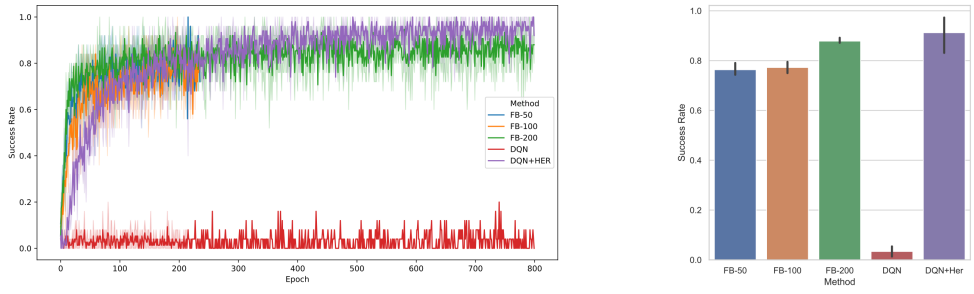

Figure E.4: Ms. Pacman: Comparative performance of FB for different dimensions and DQN. **Left**: the success rate averaged over 20 randomly selected goals as function of the training epochs. **Right**: the success rate averaged over the 184 handcrafted goals after training epochs. Note that FB-50 and F-100 have been trained only for 200 epochs.

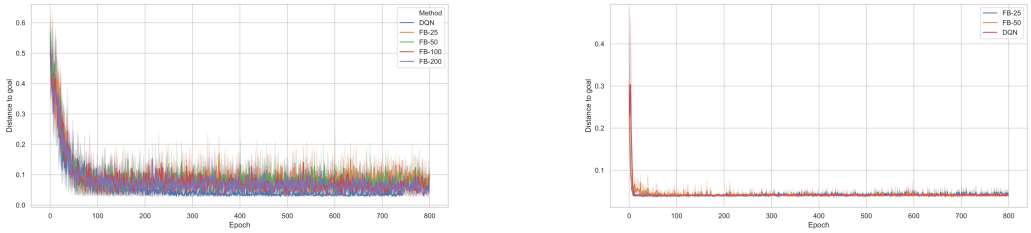

Figure E.5: Distance to goal of FB for different dimensions and DQN as function of training epochs. **Left**: Continuous maze. **Right**: FetchReach.

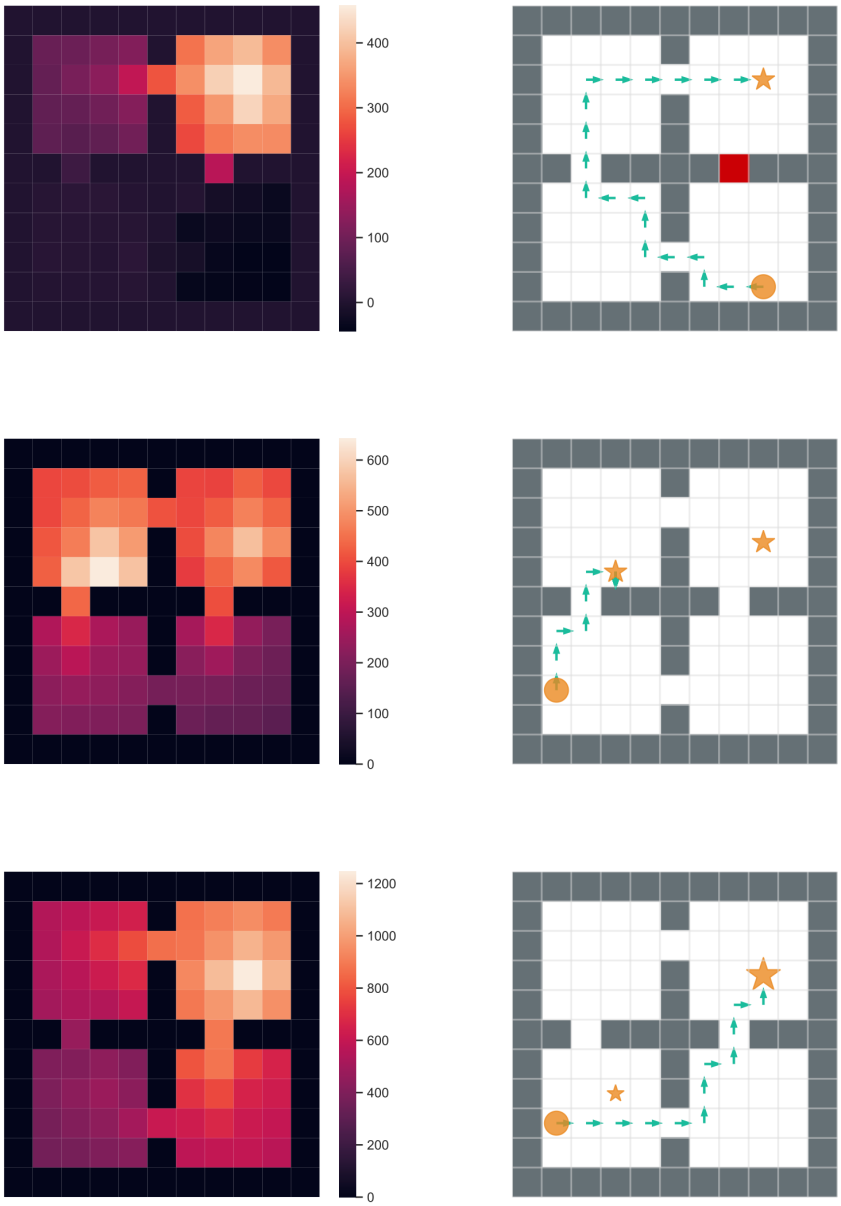

Figure E.6: Discrete Maze: Heatmap plots of $\max_{a \in A} F(s, a, z_R)^\top z_R$ (left) and trajectories of the Boltzmann policy with respect to $F(s, a, z_R)^\top z_R$ with temperature $\tau = 1$ (right). **Top row**: for the task of reaching a target while avoiding a forbidden region, **Middle row**: for the task of reaching the closest goal among two equally rewarding positions, **Bottom row**: choosing between a small, close reward and a large, distant one.

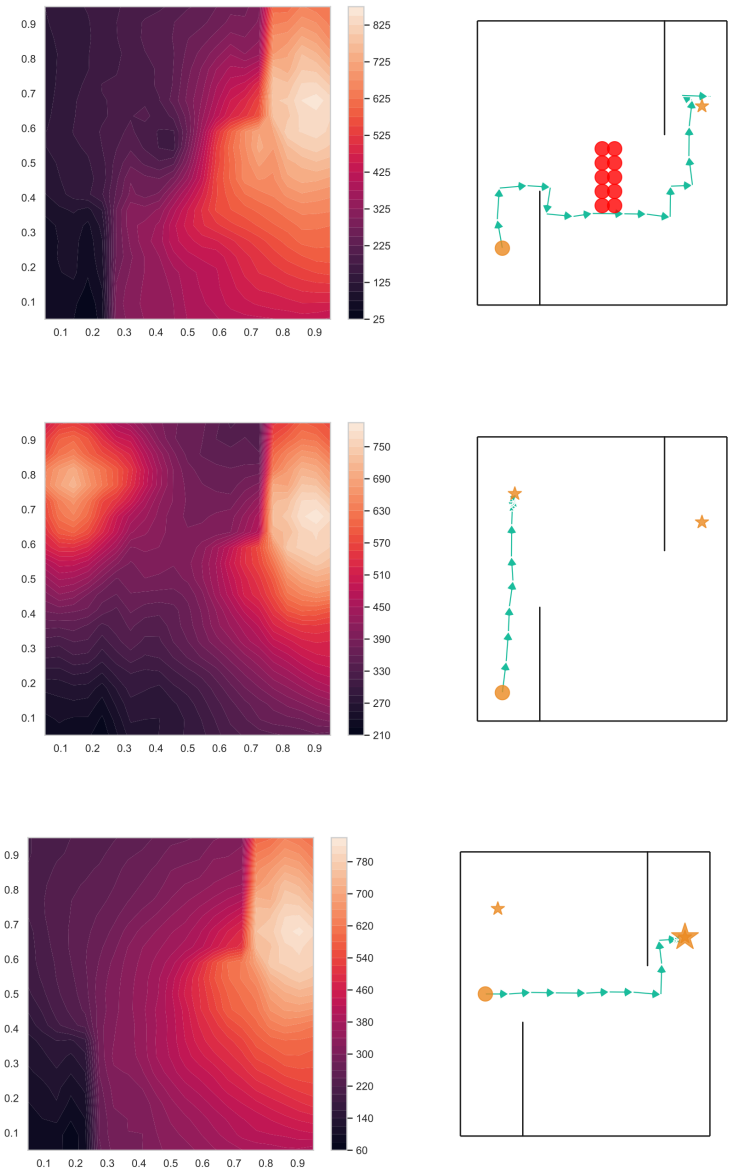

Figure E.7: Continuous Maze: Contour plots plot of $\max_{a \in A} F(s, a, z_R)^\top z_R$ (left) and trajectories of the ε greedy policy with respect to $F(s, a, z_R)^\top z_R$ with $\varepsilon = 0.1$ (right). **Left**: for the task of reaching a target while avoiding a forbidden region, **Middle**: for the task of reaching the closest goal among two equally rewarding positions, **Right**: choosing between a small, close reward and a large, distant one..

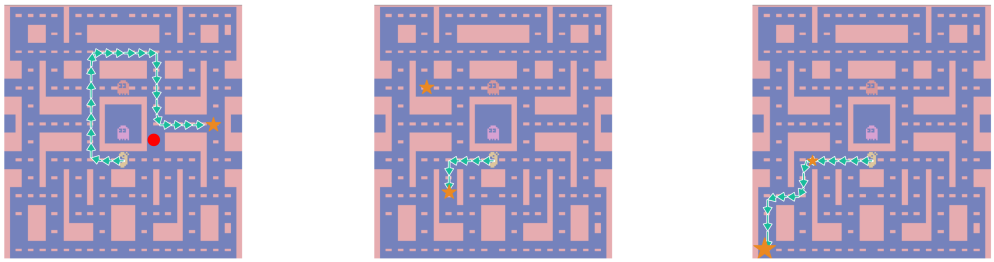

Figure E.8: Ms. Pacman: Trajectories of the ε greedy policy with respect to $F(s, a, z_R)^\top z_R$ with $\varepsilon = 0.1$ (right). **Top row**: for the task of reaching a target while avoiding a forbidden region, **Middle row**: for the task of reaching the closest goal among two equally rewarding positions, **Bottom row**: choosing between a small, close reward and a large, distant one..

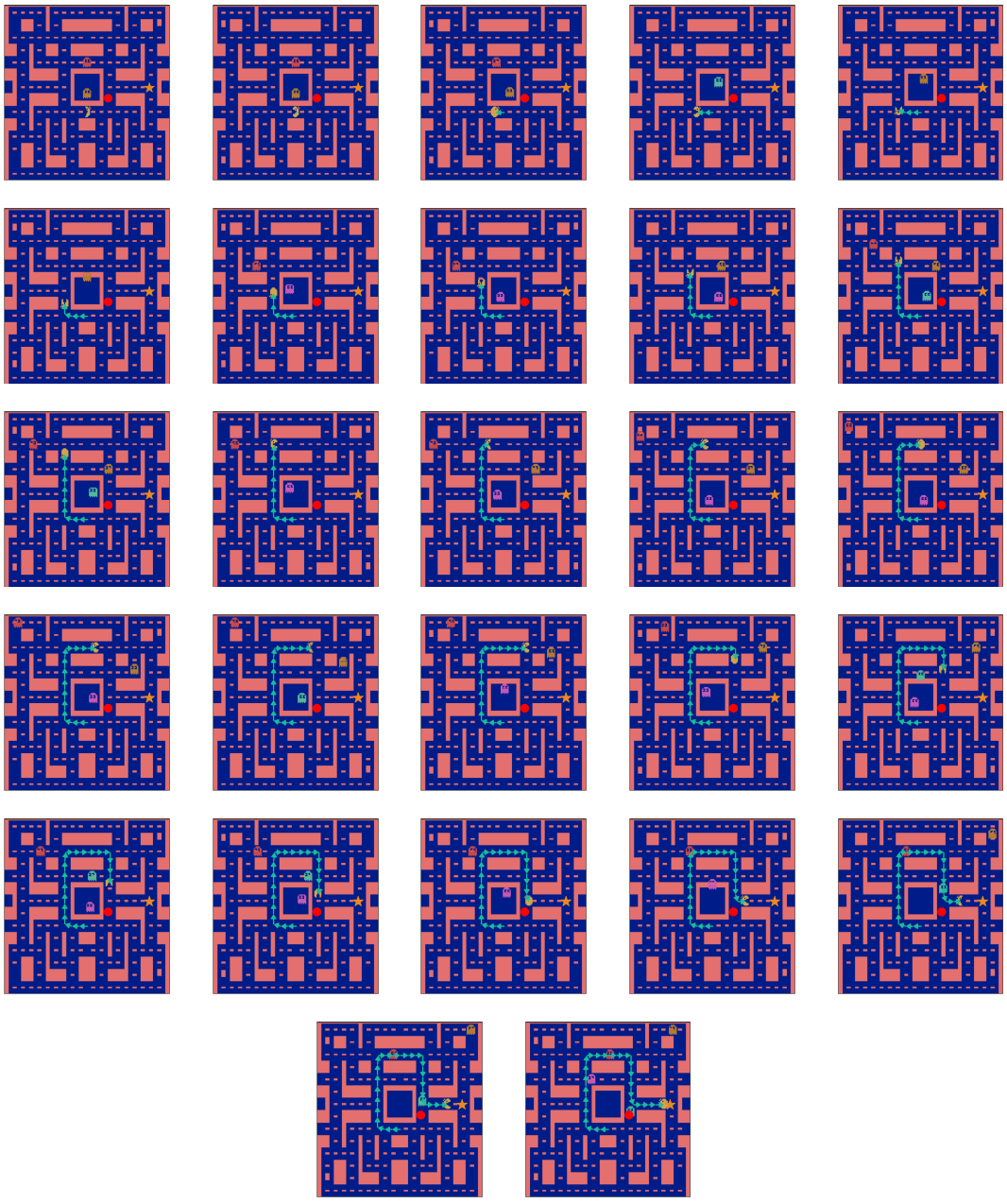

Figure E.9: Full series of frames in Ms. Pacman along the trajectory generated by the $F^{\top}B$ policy for the task of reaching a target position (star shape ☆) while avoiding forbidden positions (red circle).

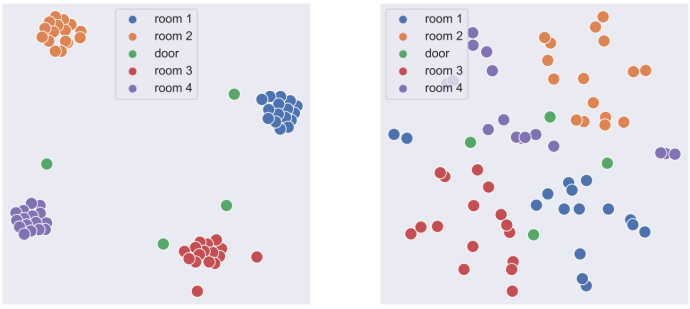

Figure E.10: Discrete maze: Visualization of FB embedding vectors after projecting them in two-dimensional space with t-SNE. **Left**: the F embedding for $z = 0$. **Right**: the B embedding. Note how both embeddings recover the foor-room and door structure of the original environment. The spread of B embedding is due to the regularization that makes B closer to orthonormal.

Figure E.11: Continuous maze: Visualization of FB embedding vectors after projecting them in two-dimensional space with t-SNE. **Left**: the states to be mapped. **Middle**: the F embedding. **Right**: the B embedding.

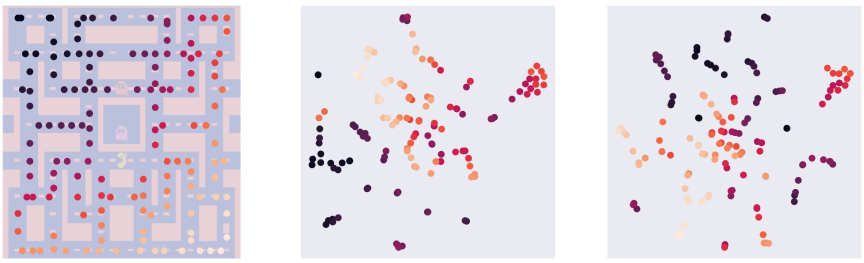

Figure E.12: Ms. Pacman: Visualization of FB embedding vectors after projecting them in two-dimensional space with t-SNE. **Left**: the agent's position corresponding to the state to be mapped. **Middle**: the F embedding for $z = 0$. **Right**: the B embedding. Note how both embeddings recover the cycle structure of the environment. F acts on visual inputs and B acts on the agent's position.

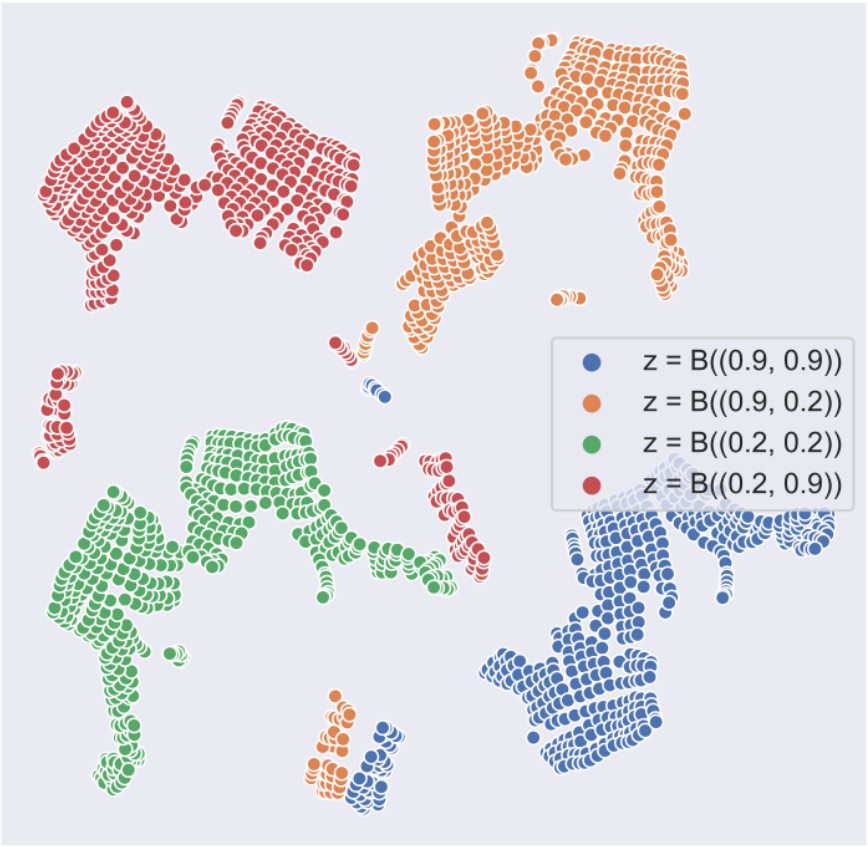

Figure E.13: Continuous maze: visualization of F embedding vectors for different z vectors, after projecting them in two-dimensional space with t-SNE.

Bibliography

Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

A. Abdolmaleki, J. T. Springenberg, Y. Tassa, R. Munos, N. Heess, and M. A. Riedmiller. Maximum a posteriori policy optimisation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=S1ANxQW0b.

J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 22–31. JMLR. org, 2017.

A. Anand, E. Racah, S. Ozair, Y. Bengio, M.-A. Côté, and R. D. Hjelm. Unsupervised state representation learning in atari. *arXiv preprint arXiv:1906.08226*, 2019.

M. Andrychowicz, D. Crow, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba. Hindsight experience replay. In *NIPS*, 2017.

K. Asadi, D. Misra, and M. L. Littman. Lipschitz continuity in model-based reinforcement learning. *arXiv preprint arXiv:1804.07193*, 2018.

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

A. Ayoub, Z. Jia, C. Szepesvari, M. Wang, and L. Yang. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pages 463–474. PMLR, 2020.

M. G. Azar, V. Gómez, and H. J. Kappen. Dynamic policy programming. *Journal of Machine Learning Research*, 13(Nov):3207–3245, 2012.

M. G. Azar, A. Lazaric, and E. Brunskill. Online stochastic optimization under correlated bandit feedback. In *ICML*, pages 1557–1565, 2014.

M. G. Azar, I. Osband, and R. Munos. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 263–272. JMLR. org, 2017.

L. Baird et al. Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the twelfth international conference on machine learning*, 1995.

A. Barreto, W. Dabney, R. Munos, J. J. Hunt, T. Schaul, D. Silver, and H. P. van Hasselt. Successor features for transfer in reinforcement learning. In *NIPS*, 2017.

M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *J. Artif. Int. Res.*, 47 (1):253–279, May 2013a. ISSN 1076-9757.

M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013b.

R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

M. Benzi and V. Simoncini. On the eigenvalues of a class of saddle point matrices. *Numerische Mathematik*, 2006.

D. P. Bertsekas. Temporal difference methods for general projected equations. *IEEE Transactions on Automatic Control*, 2011.

D. P. Bertsekas and J. N. Tsitsiklis. Neuro-dynamic programming: an overview. In *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*. IEEE, 1995.

L. Blier, C. Tallec, and Y. Ollivier. Learning successor states and goal-dependent values: A mathematical viewpoint. *arXiv preprint arXiv:2101.07123*, 2021.

V. I. Bogachev. *Measure theory*. Springer, 2007.

V. S. Borkar and S. P. Meyn. The o.d.e. method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, jan 2000.

D. Borsa, A. Barreto, J. Quan, D. Mankowitz, R. Munos, H. van Hasselt, D. Silver, and T. Schaul. Universal successor features approximators. *arXiv preprint arXiv:1812.07626*, 2018.

P. Brémaud. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*, volume 31. 1999.

G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym, 2016.

S. Bubeck, G. Stoltz, C. Szepesvári, and R. Munos. Online optimization in x-armed bandits. In *Advances in Neural Information Processing Systems*, pages 201–208, 2009.

Q. Cai, Z. Yang, C. Jin, and Z. Wang. Provably efficient exploration in policy optimization. *arXiv preprint arXiv:1912.05830*, 2019.

T. Cao and A. Krishnamurthy. Provably adaptive reinforcement learning in metric spaces. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, 2020.

J. Chen and N. Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, pages 1042–1051. PMLR, 2019.

Y. Chen, G. Lan, and Y. Ouyang. Optimal primal-dual methods for a class of saddle point problems. *SIAM Journal on Optimization*, 2014.

W. C. Cheung, D. Simchi-Levi, and R. Zhu. Learning to optimize under non-stationarity. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1079–1087, 2019.

W. C. Cheung, D. Simchi-Levi, and R. Zhu. Reinforcement learning for non-stationary markov decision processes: The blessing of (more) optimism. *arXiv preprint arXiv:2006.14389*, 2020.

Y. Chow and H. Teicher. Probability theory: Independence, interchangeability, martingales. *Journal of the American Statistical Association*, 93, 12 1998. doi: 10.2307/2670078.

B. Dai, A. Shaw, L. Li, L. Xiao, N. He, Z. Liu, J. Chen, and L. Song. Sbeed: Convergent reinforcement learning with nonlinear function approximation. In *International Conference on Machine Learning*, pages 1125–1134, 2018.

G. Dalal, B. Szorenyi, G. Thoppe, and S. Mannor. Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning. *arXiv preprint arXiv:1703.05376*, 2017.

C. Dann, T. Lattimore, and E. Brunskill. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. In *NIPS*, 2017.

P. Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993.

A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, 2014.

O. D. Domingues, P. Ménard, M. Pirotta, E. Kaufmann, and M. Valko. A kernel-based approach to non-stationary reinforcement learning in metric spaces. *arXiv preprint arXiv:2007.05078*, 2020a.

O. D. Domingues, P. Ménard, M. Pirotta, E. Kaufmann, and M. Valko. Regret bounds for kernel-based reinforcement learning. *arXiv preprint arXiv:2004.05599*, 2020b.

K. Dong, J. Peng, Y. Wang, and Y. Zhou. Root-n-regret for learning in markov decision processes with function approximation and low bellman rank. In *Conference on Learning Theory*, pages 1554–1557. PMLR, 2020.

M. D. Donsker and S. S. Varadhan. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on Pure and Applied Mathematics*, 36(2):183–212, 1983.

M. P. Drazin. Pseudo-inverses in associative rings and semigroups. *The American Mathematical Monthly*, aug 1958.

S. S. Du, J. Chen, L. Li, L. Xiao, and D. Zhou. Stochastic variance reduction methods for policy evaluation. In *International Conference on Machine Learning*, 2017.

S. S. Du, J. D. Lee, G. Mahajan, and R. Wang. Agnostic q-learning with function approximation in deterministic systems: Near-optimal bounds on approximation error and sample complexity. *Advances in Neural Information Processing Systems*, 33, 2020.

L. Faury, Y. Russac, M. Abeille, and C. Calauzènes. A technical note on non-stationary parametric bandits: Existing mistakes and preliminary solutions. In *ALT*, 2021.

N. Ferns, P. Panangaden, and D. Precup. Metrics for finite markov decision processes. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 162–169. AUAI Press, 2004.

D. Foster and P. Dayan. Structure in the space of value functions. *Machine Learning*, 49(2):325–346, 2002.

P. Gajane, R. Ortner, and P. Auer. A sliding-window algorithm for markov decision processes with arbitrarily changing rewards and transitions. *arXiv preprint arXiv:1805.10066*, 2018.

M. Geist, B. Scherrer, and O. Pietquin. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169, 2019.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

C. Grimm, I. Higgins, A. Barreto, D. Teplyashin, M. Wulfmeier, T. Hertweck, R. Hadsell, and S. Singh. Disentangled cumulants help successor representations transfer to new tasks. *arXiv preprint arXiv:1911.10866*, 2019.

C. M. Grinstead and J. L. Snell. *Introduction to probability*. American Mathematical Soc., 1997.

A. Gruslys, W. Dabney, M. G. Azar, B. Piot, M. Bellemare, and R. Munos. The reactor: A fast and sample-efficient actor-critic agent for reinforcement learning. *arXiv preprint arXiv:1704.04651*, 2017.

T. Haarnoja, H. Tang, P. Abbeel, and S. Levine. Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1352–1361. JMLR. org, 2017.

T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870, 2018.

S. Hansen, W. Dabney, A. Barreto, T. Van de Wiele, D. Warde-Farley, and V. Mnih. Fast task inference with variational intrinsic successor features. *arXiv preprint arXiv:1906.05030*, 2019.

A. Harutyunyan, M. G. Bellemare, T. Stepleton, and R. Munos. Q (λ) with off-policy corrections. In *International Conference on Algorithmic Learning Theory*. Springer, 2016.

C.-W. Huang, A. Touati, L. Dinh, M. Drozdzal, M. Havaei, L. Charlin, and A. C. Courville. Learnable explicit density for continuous latent space and variational inference. *ArXiv*, abs/1710.02248, 2017a.

C.-W. Huang, A. Touati, P. Vincent, G. K. Dziugaite, A. Lacoste, and A. Courville. Stochastic neural network with kronecker flow. In *International Conference on Artificial Intelligence and Statistics*, pages 4184–4194. PMLR, 2020.

G. Huang, H. Berard, A. Touati, G. Gidel, P. Vincent, and S. Lacoste-Julien. Parametric adversarial divergences are good task losses for generative modeling. *arXiv preprint arXiv:1708.02511*, 2017b.

T. Jaakkola, M. I. Jordan, and S. P. Singh. On the convergence of stochastic iterative dynamic programming algorithms. *Neural computation*, 6(6):1185–1201, 1994.

T. Jaksch, R. Ortner, and P. Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 2010.

Q. Jian, R. Fruit, M. Pirotta, and A. Lazaric. Exploration bonus for regret minimization in discrete and continuous average reward mdps. In *Advances in Neural Information Processing Systems*, pages 4891–4900, 2019.

N. Jiang, A. Krishnamurthy, A. Agarwal, J. Langford, and R. E. Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2017.

C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan. Is q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pages 4863–4873, 2018.

C. Jin, A. Krishnamurthy, M. Simchowitz, and T. Yu. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pages 4870–4879. PMLR, 2020a.

C. Jin, Z. Yang, Z. Wang, and M. I. Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020b.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, 2013.

S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, pages 267–274, 2002.

S. Kakade, M. J. Kearns, and J. Langford. Exploration in metric state spaces. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 306–312, 2003.

S. M. Kakade. *On the sample complexity of reinforcement learning*. PhD thesis, UCL (University College London), 2003.

N. R. Ke, A. Singh, A. Touati, A. Goyal, Y. Bengio, D. Parikh, and D. Batra. Modeling the long term future in model-based reinforcement learning. In *International Conference on Learning Representations*, 2018.

N. R. Ke, A. Singh, A. Touati, A. Goyal, Y. Bengio, D. Parikh, and D. Batra. Modeling the long term future in model-based reinforcement learning. In *ICLR*, 2019.

M. Kearns and S. Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 2002.

J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. Van Nostrand, New York, 1960.

B. Kim and A. Tewari. Randomized exploration for non-stationary stochastic linear bandits. In *Conference on Uncertainty in Artificial Intelligence*, pages 71–80. PMLR, 2020.

R. Kleinberg, A. Slivkins, and E. Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 681–690. ACM, 2008.

N. Kodali, J. Abernethy, J. Hays, and Z. Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

V. R. Konda and J. N. Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014. Citeseer, 2000.

I. Kostrikov. Pytorch implementations of reinforcement learning algorithms. https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail, 2018.

I. Kostrikov, O. Nachum, and J. Tompson. Imitation learning via off-policy distribution matching. *arXiv preprint arXiv:1912.05032*, 2019.

K. Lakshmanan, R. Ortner, and D. Ryabko. Improved regret bounds for undiscounted continuous reinforcement learning. In *International Conference on Machine Learning*, pages 524–532, 2015.

C. Lakshminarayanan and C. Szepesvári. Linear stochastic approximation: Constant step-size and iterate averaging. *arXiv preprint arXiv:1709.04073*, 2017.

T. Lattimore, M. Hutter, P. Sunehag, et al. The sample-complexity of general reinforcement learning. In *Proceedings of the 30th International Conference on Machine Learning*. Journal of Machine Learning Research, 2013.

S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

L.-J. Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, may 1992.

B. Liu, J. Liu, M. Ghavamzadeh, S. Mahadevan, and M. Petrik. Finite-sample analysis of proximal gradient td algorithms. In *UAI*. Citeseer, 2015.

S. Liu and H. Su. Regret bounds for discounted mdps. *arXiv preprint arXiv:2002.05138*, 2020.

Y. Liu, A. Swaminathan, A. Agarwal, and E. Brunskill. Provably good batch reinforcement learning without great exploration. *arXiv preprint arXiv:2007.08202*, 2020.

S. V. Macua, J. Chen, S. Zazo, and A. H. Sayed. Distributed policy evaluation under multiple behavior strategies. *IEEE Transactions on Automatic Control*, 2015.

H. R. Maei. Gradient temporal-difference learning algorithms. 2011.

H. R. Maei and R. S. Sutton. Gq (λ): A general gradient algorithm for temporal-difference prediction learning with eligibility traces. In *Proceedings of the Third Conference on Artificial General Intelligence*, 2010.

S. Mahadevan and M. Maggioni. Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *Journal of Machine Learning Research*, 8(10), 2007.

A. R. Mahmood, H. Yu, and R. S. Sutton. Multi-step off-policy learning without importance sampling ratios. *arXiv preprint arXiv:1702.03006*, 2017.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, 2016.

R. Munos. Error bounds for approximate value iteration. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 1006. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

R. Munos and C. Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(5), 2008.

R. Munos, T. Stepleton, A. Harutyunyan, and M. Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, 2016.

R. Munos et al. From bandits to monte-carlo tree search: The optimistic principle applied to optimization and planning. *Foundations and Trends® in Machine Learning*, 7(1):1–129, 2014.

O. Nachum, Y. Chow, B. Dai, and L. Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. In *Advances in Neural Information Processing Systems*, pages 2315–2325, 2019a.

O. Nachum, B. Dai, I. Kostrikov, Y. Chow, L. Li, and D. Schuurmans. Algaedice: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*, 2019b.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 2009.

G. Neu and J. Olkhovskaya. Online learning in mdps with linear function approximation and bandit feedback. *arXiv preprint arXiv:2007.01612*, 2020.

G. Neu, A. Jonsson, and V. Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

X. Nguyen, M. J. Wainwright, M. I. Jordan, et al. On surrogate loss functions and f-divergences. *The Annals of Statistics*, 37(2):876–904, 2009.

R. Ortner. Pseudometrics for state aggregation in average reward markov decision processes. In *International Conference on Algorithmic Learning Theory*, pages 373–387. Springer, 2007.

R. Ortner and D. Ryabko. Online regret bounds for undiscounted continuous reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1763–1771, 2012.

R. Ortner, P. Gajane, and P. Auer. Variational regret bounds for reinforcement learning. In *UAI*, page 16, 2019.

I. Osband, D. Russo, and B. Van Roy. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, 2013.

I. Osband, B. Van Roy, D. J. Russo, and Z. Wen. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62, 2019.

B. Palaniappan and F. Bach. Stochastic variance reduction methods for saddle-point problems. In *Advances in Neural Information Processing Systems*, 2016.

J. Pazis and R. Parr. Pac optimal exploration in continuous space markov decision processes. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

Z. Peng, A. Touati, P. Vincent, and D. Precup. Svrg for policy evaluation with fewer gradient evaluations. *arXiv preprint arXiv:1906.03704*, 2019.

M. Pirotta, M. Restelli, A. Pecorino, and D. Calandriello. Safe policy iteration. In *International Conference on Machine Learning*, pages 307–315, 2013.

M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

D. Pollard. Empirical processes: theory and applications. In *NSF-CBMS regional conference series in probability and statistics*, pages i–86. JSTOR, 1990.

D. Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, 2000.

D. Precup, R. S. Sutton, and S. Dasgupta. Off-policy temporal difference learning with function approximation. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, 2001.

M. L. Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994. ISBN 0471619779.

J. Qian, R. Fruit, M. Pirotta, and A. Lazaric. Exploration bonus for regret minimization in undiscounted discrete and continuous markov decision processes. *arXiv preprint arXiv:1812.04363*, 2018.

P. Rauber, A. Ummadisingu, F. Mutz, and J. Schmidhuber. Hindsight policy gradients. In *International Conference on Learning Representations*, 2018.

N. Roese. Counterfactual thinking and decision making. *Psychonomic bulletin & review*, 6(4):570–578, 1999.

J. Romoff, P. Henderson, A. Touati, E. Brunskill, J. Pineau, and Y. Ollivier. Separating value functions across time-scales. In *International Conference on Machine Learning*, pages 5468–5477. PMLR, 2019.

J. Romoff, P. Henderson, D. Kanaa, E. Bengio, A. Touati, P.-L. Bacon, and J. Pineau. Tdprop: Does adaptive optimization with jacobi preconditioning help temporal difference learning? In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1082–1090, 2021.

L. Rosasco, S. Villa, and B. C. Vũ. Stochastic forward–backward splitting for monotone inclusions. *Journal of Optimization Theory and Applications*, 2016.

Y. Russac, C. Vernade, and O. Cappé. Weighted linear bandits for non-stationary environments. In *Advances in Neural Information Processing Systems*, pages 12017–12026, 2019.

T. Schaul, D. Horgan, K. Gregor, and D. Silver. Universal value function approximators. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1312–1320, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/schaul15.html.

J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

S. E. Shreve and D. P. Bertsekas. Alternative theoretical frameworks for finite horizon discrete-time stochastic optimal control. *SIAM Journal on control and optimization*, 16(6):953–978, 1978.

S. Sinclair, T. Wang, G. Jain, S. Banerjee, and C. Yu. Adaptive discretization for model-based reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020.

S. R. Sinclair, S. Banerjee, and C. L. Yu. Adaptive discretization for episodic reinforcement learning in metric spaces. *arXiv preprint arXiv:1910.08151*, 2019.

A. Slivkins. Contextual bandits with similarity information. *The Journal of Machine Learning Research*, 15(1):2533–2568, 2014.

Z. Song and W. Sun. Efficient model-free reinforcement learning in metric spaces. *arXiv preprint arXiv:1905.00475*, 2019.

B. K. Sriperumbudur, K. Fukumizu, A. Gretton, B. Schölkopf, and G. R. Lanckriet. On integral probability metrics,ϕ-divergences and binary classification. *arXiv preprint arXiv:0901.2698*, 2009.

K. L. Stachenfeld, M. M. Botvinick, and S. J. Gershman. The hippocampus as a predictive map. *Nature neuroscience*, 20(11):1643, 2017.

A. L. Strehl and M. L. Littman. A theoretical analysis of model-based interval estimation. In *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005.

M. Strens. A bayesian framework for reinforcement learning. In *ICML*, 2000.

R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 1988.

R. S. Sutton. Introduction to reinforcement learning with function approximation. Tutorial Session of the Neural Information Processing Systems Conference, 2015.

R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998. ISBN 0262193981.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 2nd edition, Near-final draft – May 27, 2018.

R. S. Sutton and B. Tanner. Temporal-difference networks. In *Advances in Neural Information Processing Systems 17*, 2004.

R. S. Sutton, D. A. McAllester, S. P. Singh, Y. Mansour, et al. Policy gradient methods for reinforcement learning with function approximation. In *NIPs*, volume 99, pages 1057–1063. Citeseer, 1999a.

R. S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999b.

R. S. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, aug 1999c.

R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML*. ACM Press, 2009a.

R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009b.

R. S. Sutton, H. R. Maei, and C. Szepesvári. A convergent $o(n)$ temporal-difference algorithm for off-policy learning with linear function approximation. In *Advances in neural information processing systems*, 2009c.

R. S. Sutton, J. Modayil, M. Delp, T. Degris, P. M. Pilarski, A. White, and D. Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '11, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems.

R. S. Sutton, A. R. Mahmood, and M. White. An emphatic approach to the problem of off-policy temporal-difference learning. *The Journal of Machine Learning Research*, 2015.

E. Talvitie. Self-correcting models for model-based reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 1933.

A. Touati and Y. Ollivier. Learning one representation to optimize all rewards. *arXiv preprint arXiv:2103.07945*, 2021.

A. Touati and P. Vincent. Efficient learning in non-stationary linear markov decision processes. *arXiv preprint arXiv:2010.12870*, 2020a.

A. Touati and P. Vincent. Sharp analysis of smoothed bellman error embedding. *arXiv preprint arXiv:2007.03749*, 2020b.

A. Touati, P.-L. Bacon, D. Precup, and P. Vincent. Convergent tree backup and retrace with function approximation. In *International Conference on Machine Learning*, pages 4955–4964. PMLR, 2018.

A. Touati, H. Satija, J. Romoff, J. Pineau, and P. Vincent. Randomized value functions via multiplicative normalizing flows. In *Uncertainty in Artificial Intelligence*, pages 422–432. PMLR, 2020a.

A. Touati, A. A. Taiga, and M. G. Bellemare. Zooming for efficient model-free reinforcement learning in metric spaces. *arXiv preprint arXiv:2003.04069*, 2020b.

A. Touati, A. Zhang, J. Pineau, and P. Vincent. Stable policy optimization via off-policy divergence regularization. In *Conference on Uncertainty in Artificial Intelligence*, pages 1328–1337. PMLR, 2020c.

J. N. Tsitsiklis, B. Van Roy, et al. An analysis of temporal-difference learning with function approximation. *IEEE transactions on automatic control*, 1997.

L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

H. van Hasselt, A. R. Mahmood, and R. S. Sutton. Off-policy td (λ) with a true online equivalence. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence, Quebec City, Canada*, 2014.

N. Vieillard, O. Pietquin, and M. Geist. Deep conservative policy iteration. *arXiv preprint arXiv:1906.09784*, 2019.

H.-T. Wai, M. Hong, Z. Yang, Z. Wang, and K. Tang. Variance reduced policy evaluation with smooth function approximation. *Advances in Neural Information Processing Systems*, 32:5784–5795, 2019.

M. Wang and D. P. Bertsekas. Stabilization of stochastic iterative methods for singular and nearly singular linear systems. *Mathematics of Operations Research*, 2013.

Q. Wang, Y. Li, J. Xiong, and T. Zhang. Divergence-augmented policy optimization. In *Advances in Neural Information Processing Systems*, pages 6097–6108, 2019a.

R. Wang, R. R. Salakhutdinov, and L. Yang. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. *Advances in Neural Information Processing Systems*, 33, 2020.

Y. Wang, W. Chen, Y. Liu, Z.-M. Ma, and T.-Y. Liu. Finite sample analysis of the gtd policy evaluation algorithms in markov setting. In *Advances in Neural Information Processing Systems*, pages 5510–5519, 2017.

Y. Wang, H. He, and X. Tan. Truly proximal policy optimization. *arXiv preprint arXiv:1903.07940*, 2019b.

Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu, and N. de Freitas. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*, 2016.

C. J. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

C.-Y. Wei and H. Luo. Non-stationary reinforcement learning without prior knowledge: An optimal black-box approach. In *COLT*, 2021.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

L. F. Yang and M. Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. *arXiv preprint arXiv:1905.10389*, 2019.

L. F. Yang, C. Ni, and M. Wang. Learning to control in metric space with optimal regret. *arXiv preprint arXiv:1905.01576*, 2019.

A. Zanette, D. Brandfonbrener, E. Brunskill, M. Pirotta, and A. Lazaric. Frequentist regret bounds for randomized least-squares value iteration. In *International Conference on Artificial Intelligence and Statistics*, pages 1954–1964, 2020a.

A. Zanette, A. Lazaric, M. Kochenderfer, and E. Brunskill. Learning near optimal policies with low inherent bellman error. *arXiv preprint arXiv:2003.00153*, 2020b.

J. Zhang, J. T. Springenberg, J. Boedecker, and W. Burgard. Deep reinforcement learning with successor features for navigation across similar environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2371–2378. IEEE, 2017.

P. Zhao and L. Zhang. Non-stationary linear bandits revisited. *arXiv preprint arXiv:2103.05324*, 2021.

P. Zhao, L. Zhang, Y. Jiang, and Z.-H. Zhou. A simple approach for non-stationary linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 746–755. PMLR, 2020.

H. Zhou, J. Chen, L. R. Varshney, and A. Jagmohan. Nonstationary reinforcement learning with linear function approximation. *arXiv preprint arXiv:2010.04244*, 2020.

X. Zhu and D. Dunson. Stochastic lipschitz q-learning. 2019.