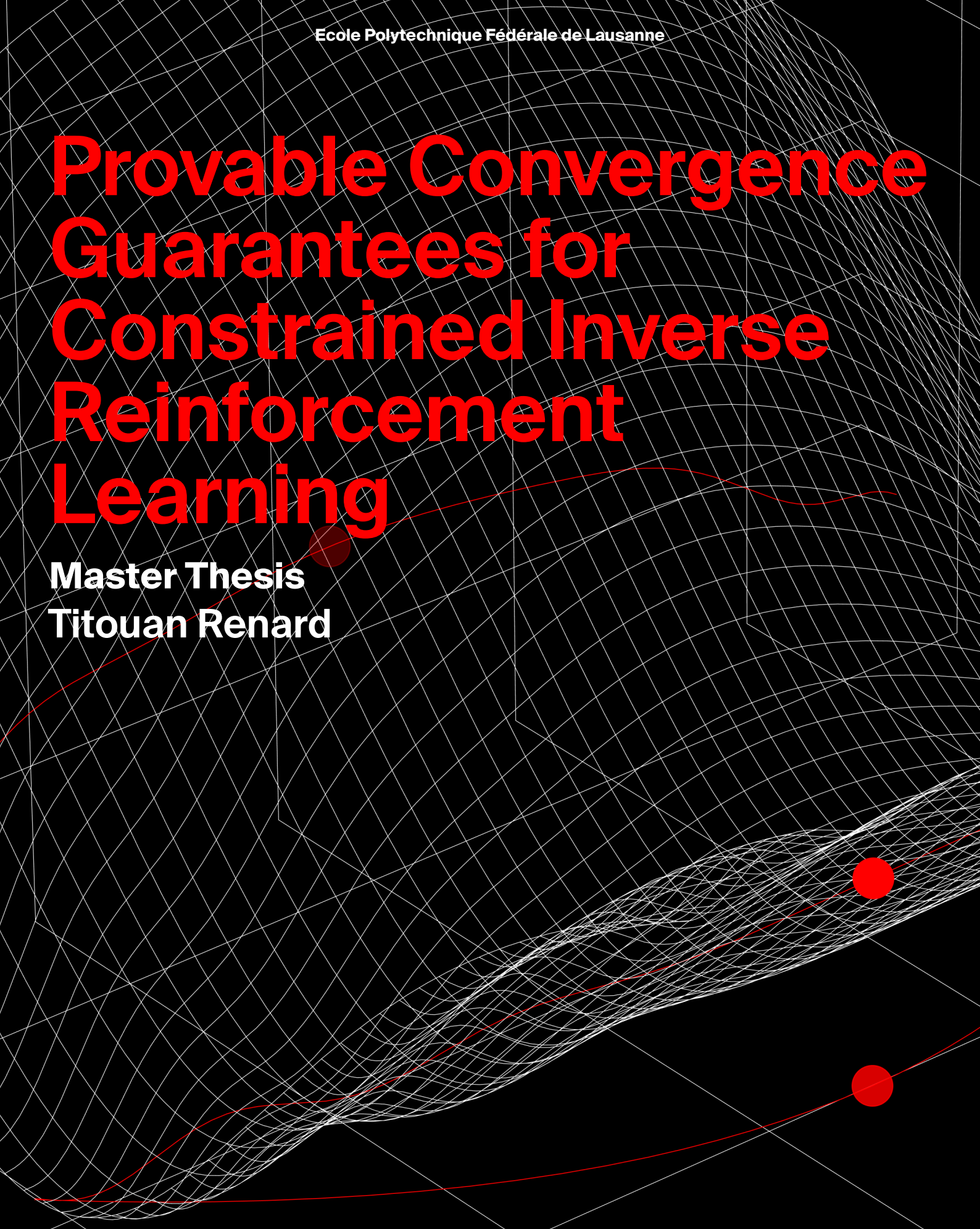
Ecole Polytechnique Fédérale de Lausanne

Provable Convergence Guarantees for Constrained Inverse Reinforcement Learning

Master Thesis

Titouan Renard



EPFL

Provable Convergence Guarantees for Constrained Inverse Reinforcement Learning

by

Titouan Renard

to obtain the degree of Master of Science
at EPFL,
to be defended publicly on Friday July 14, 2023.

Student number: 272257
Project duration: March 27, 2023 – June 30, 2023

Prof. Maryam Kamgarpour,	EPFL, supervisor
Andreas Schlaginhaufen,	EPFL, Assistant
Tingting Ni,	EPFL, Assistant
Anna Maria Maddux,	EPFL, Assistant
Dr. Umsanova Ilnura,	ETH, External Expert

Cover: Illustration of the loss landscape of the CIRL Lagrangian by Titouan Renard
Style: EPFL Report Style, with modifications by Batuhan Faik Derinbay

An electronic version of this thesis is available at `https://www.epfl.ch/`.

EPFL

Abstract

By incorporating known constraints into the inverse reinforcement learning (*IRL*) framework, constrained inverse reinforcement learning (*CIRL*) can learn behaviors from expert demonstration while satisfying a set of pre-defined constraints. This makes *CIRL* relevant in safety-critical domains, as it provides a direct way to devise AI systems that enforce safety requirements. This master thesis proposes and analyzes an algorithm, termed *NPG-CIRL*, that solves the problem of *CIRL*. Our algorithm implements a primal-dual scheme that extends the natural policy gradient (*NPG*) algorithm to the *CIRL* setting. We provide a finite-time analysis of the algorithm's global convergence in the idealized exact gradient setting and the more practical stochastic gradient setting. We show that the algorithm requires $O(1/\epsilon^2)$ gradient evaluations to reach an ϵ-approximate solution and to satisfy the imposed constraints. Our analysis also quantifies the sample complexity, showing that the algorithm requires $O(1/\epsilon^4)$ samples to achieve convergence when using Monte Carlo gradient estimation techniques.

Acknowledgements

I would like to thank Pr. Maryam Kamgarpour from the SYCAMORE lab at EPFL for supervising me during this project and allowing me to spend time doing research in the amazing environment that SYCAMORE has offered me.

I want to express my gratitude to the three PhD advisors I was lucky to have to supervise, advise and support me during my time at SYCAMORE. Anna Maddux for her excellent writing advice and for the many times she showed me how to move from a vague intuition to a valuable, usable result. Tingting Ni for sharing her knowledge of policy gradient algorithms and her sharp proofreading that highlighted many inconsistencies I would have missed without her. Andreas Schlaginhaufen for sharing his deep understanding of the *CIRL* problem and for all the time he devoted to helping me in my work.

I would also like to thank my friends and family that helped and supported me during this time. A special thanks must go to Gabriel Vallat for making me aware, during his master thesis at SYCAMORE, of the research being done in the lab, without whom I would not have even applied for a master thesis here. I am also grateful to Paul, Lucie, Dana and Elise, the friends with whom I have shared my office and spent coffee breaks with when working on my thesis. They have made the experience a lot more fun than it would have been otherwise.

Titouan Renard
EPFL, July 2023

Notation

In the following text, we use the following conventions for notation. We denote all of our vectors $\boldsymbol{x}$ in bold, our scalars x in regular weight and our matrices X are capitalized. When used between two vectors $\boldsymbol{x} > \boldsymbol{y}$, the comparison is element-wise. The transpose operation is denoted $X^\top$. The scalar product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ between two vectors $\boldsymbol{x}$, $\boldsymbol{y}$ is denoted using angled brackets. When a single-variable function $f : \mathbb{R} \to \mathbb{R}$ is applied to a vector $f(\boldsymbol{x})$, it is applied element-wise.

We write $\|\boldsymbol{x}\|_p$ to denote the p norm of a vector $\boldsymbol{x}$. The notation $\|X\|$ denotes the spectral norm of a matrix. The $\|X\|_p$ notation denotes the operator norm of a matrix induced by the vector norm p. The $\|X\|_F$ notation denotes the Frobenius norm over the matrix X.

We write $|S|$ for the cardinality (the number of elements) of a discrete set S. When considering a discrete function $f : S \to \mathbb{R}$, we write $f(s)$ for a specific element $s \in S$. We also use the following shorthand to think of discrete functions, we let the vector $\boldsymbol{f} = [f(s_1), ..., f(x_{|X|})]^\top \in \mathbb{R}^{|X|}$ contain all of the values taken by the function $f : S \to \mathbb{R}$ over the finite set S. When using that notation of discrete functions *as vectors*, we write the function $\boldsymbol{f}$ in bold.

We write Δ_S for the probability simplex over the set S. When considering functions over sets (for instance, a set of simplices), we write Δ_U^S; here, this is a mapping from a set S to a set of probability simplices over the discrete set U.

Quantities that evolve with time, such as algorithm parameters or the current state in a stochastic process, are expressed as $x^{(t)}$ for the t-th iteration (where x is the quantity).

Contents

1

Introduction

Reinforcement learning (*RL*) has emerged as a prominent field of artificial intelligence, enabling great success in various applications and industries, such as robotics [Peng et al. 2020; Lee et al. 2020], autonomous vehicle control [Kiran et al. 2022], software engineering [Mankowitz et al. 2023] and bioinformatics [Wang et al. 2022]. By allowing agents to learn optimal behaviours from trial and error, RL has the potential to enhance efficiency in many industries and address complex decision-making problems that were previously unsolvable.

However, *RL* is often called a ”black box method” because of its inherent complexity and lack of transparency. The decision-making processes and learned policies of *RL* agents are not easily explainable by humans, and most *RL* methods do not allow for explicit constraints to be specified. Thich has led to concerns regarding safety and explainability in *RL* agents.

Safe *RL* describes the research area aiming to develop algorithms and methodologies that ensure agents learn and act to minimise the risk of harmful outcomes, preventing potentially dangerous situations. This aspect is particularly crucial in domains such as healthcare and autonomous vehicles, where the impact of a wrong decision can be disastrous. As *RL* algorithms become more complex, their lack of interpretability hinders their adoption in critical domains. Explainable *RL* aims to bridge this gap by enabling humans to better understand an agent’s decisions, facilitating trust, accountability, and better integration of *RL* systems into real-world applications. Research in safe and explainable *RL* is crucial to ensure AI agents’ safe and effective deployment in real-world scenarios.

The following thesis proposes an algorithm called *NPG-CIRL* to solve the problem of *constrained inverse reinforcement learning* (*CIRL*) and provides convergence guarantees. The *CIRL* problem is an extension of the *inverse reinforcement learning* problem (*IRL*) [A. Ng, Harada, and Russell 1999], which is concerned with recovering a reward function explaining the behaviour of an expert agent. *IRL* methods are generally considered part of the broader class of imitation learning methods, which aim to enable agents to learn behaviours from demonstrations. This approach has seen great success in the field of robotics and has, for instance, been used to train locomotive policies in robotic dogs, using expert data acquired by motion capture on real canines [Peng et al. 2020]. Rewards recovered through *IRL* methods can, in turn, be used to clone the behaviour of that expert. Compared to alternative methods, such as behavioural cloning that directly tries to reproduce an expert policy with supervised learning, *IRL* presents the advantage of recovering a representation of the underlying goal of the expert. Depending on the problem structure, especially in *MDPs* with sparse rewards, the reward may provide a more compressed representation of the goal of an agent than the expert policy. Furthermore, this representation describes a goal independent of the underlying *MDP* dynamics. That makes *IRL* better suited to learn policies that are transferable across different dynamics, making *IRL* methods particularly adapted to learning behaviours that generalize across different settings. With the ability to learn transferable policies, a robotic arm could, for instance, be trained to perform a task demonstrated by a human expert, even if the kinematics of the robot are fundamentally different from those of the human demonstrator.

CIRL differs from the more extensively studied *IRL* problem by introducing known constraints into the problem formulation. The introduction of constraints guarantees that the recovered reward induces a policy that meets explicitly specified requirements. This property is crucial in safety-critical domains, such as autonomous vehicles, robotics, or medical applications. Another advantage presented when introducing known constraints into *IRL* is that *CIRL* recovers a reward that does not implicitly represent the constraints. That property implies, for instance, that a self-driving model trained on a dataset of Swiss roads could be used to learn a reward function which does not represent known constraints such as speed limits and that, when specified different restrictions, it could still meet them. For instance, the model would not require a new training dataset to learn to drive faster than the Swiss speed limitations when on the German autobahn. This would not be achievable with an unconstrained *IRL* algorithm.

These properties make *CIRL* a particularly relevant problem to study in the search for safer and more explainable AI systems. *CIRL* offers a promising approach to developing more reliable and accountable AI systems by addressing safety concerns and facilitating interpretability. These qualities are pivotal for the widespread adoption and acceptance of AI technologies in real-world applications.

1.1. Prior work

Reinforcement learning (*RL*) is a machine learning problem concerned with training agents to make sequential decisions to maximize a reward function. The problem has its roots in planning algorithms for Markov Decision processes [Bellman 1957] and in the study of reinforcement in behavioural psychology [Skinner and Ferster 1957]. *RL* methods have recently seen great success in solving complex sequential optimal decision-making problems such as the game of Go [Silver et al. 2016].

Policy gradient (*PG*) methods tackle the *RL* problem by gradient-based optimization on parameterized policies [R. J. Williams 1992; Sutton et al. 1999]. Recent works solving real-world problems such as robotic locomotion [Lee et al. 2020] or learning to play atari games [Schulman, Levine, et al. 2015; Haarnoja et al. 2018] have shown the efficiency of policy gradient algorithms, when they are used in conjunction with deep neural networks to parameterize the policies. Various modifications to the original policy gradient formulation have been proposed [Kakade 2001; Schulman, Levine, et al. 2015; Schulman, Wolski, et al. 2017]. In particular, the so-called natural policy gradient (*NPG*) method [Kakade 2001] preconditions the gradient steps with the Moore-Penrose pseudoinverse of the Fisher Information matrix, which enables the algorithm to adapt to the geometry of the problem. Trust-region policy optimization (*TRPO*), a method very closely related to *NPG*, is one of the most successful deep reinforcement learning method in practice [Schulman, Levine, et al. 2015].

It is only fairly recently that *PG* methods have been shown to converge globally. [Agarwal et al. 2020] have shown that directly parameterized, projected policy gradient ascent converges at an $O(1/\sqrt{T})$ rate and that it asymptotically converges globally with policy parameterization when given access to exact gradient Oracle. Under relative entropy regularization [Agarwal et al. 2020] also show that policy gradient and natural policy gradient converge at an $O(/\sqrt{T})$ rate and $O(1/T)$ rate, respectively (also with exact gradients).

A particularly relevant area of study in the context of this work is that of regularization in *RL*. Regularization has been investigated because of its ability to accelerate convergence [Mei et al. 2020] and as a way of inciting exploration when learning [Haarnoja et al. 2018]. The impact of regularization on *PG* methods with *softmax* parameterization has been studied by [Mei et al. 2020], who show that entropy regularization can lead to a $O(e^{-T})$ global convergence rate (with exact gradients). [Cen et al. 2021] show that the *NPG* algorithm converges at an $O(e^{-T})$ rate when entropy regularization is introduced, with exact gradients as well in a setting with bounded gradient perturbations. In the setting where gradients can only be accessed through stochastic estimators *PG* methods are proved to converge globally at a $O(1/\sqrt{T})$ [Y. Ding, Zhang, and Lavaei 2021].

Safety in Markov Decision Processes (*MDPs*), and more specifically, the setting where constraints are introduced into MDPs has been extensively studied [Altman 1999]. This foundational work has in turn been used to design reinforcement learning algorithms which are able to learn policies that are subject to safety constraints [Achiam et al. 2017]. Such algorithms are termed constrained reinforcement learning (*CRL*) algorithms. The first algorithm with provable convergence guarantees of $O(1/\sqrt{T})$ rate for *CRL* was proposed in [D. Ding et al. 2020] which relaxes the constraints and solves a Lagrangian problem with primal-dual updates. The authors propose to use natural policy gradient ascent in the primal problem while relying on projected gradient descent on the dual. A more recent contribution [Ying, Y. Ding, and Lavaei 2022] considers a *dual-descent* variation on the primal-dual formulation of [D. Ding et al. 2020], in which multiple primal steps are run for each dual step. This approach guarantees a faster convergence rate of $\tilde{O}(1/T)$ (where $\tilde{O}$ hides logarithmic factors) at the cost of a more complex implementation.

The problem of learning a reward function from a dataset of expert demonstrations, *inverse reinforcement learning*, was first introduced by [Russell 1998]. The main challenge faced when solving the *IRL* problem is degeneracy. Specifically, the set of optimal policies can be shown to be invariant under a specific class of transformations [A. Ng, Harada, and Russell 1999]. This degeneracy dramatically complicates the search for a meaningful reward. Several approaches have been proposed to overcome this limitation; this includes Bayesian approaches [Ramachandran and Amir 2007] and margin-maximization techniques [Abbeel and A. Y. Ng 2004]. In this thesis, we address the degeneracy problem of IRL via maximum causal entropy *IRL* (*MCE-IRL*) [Ziebart, Bagnell, and Dey 2010]. In *MCE-IRL*, entropy regularization is introduced into the objective function to ensure that the solution is unique and overcome the degeneracy of the problem. The maximum likelihood *IRL ML-IRL* problem, an alternative, equivalent formulation of *MCE-IRL*, can be solved through a primal-dual update scheme analogue to the one proposed for *CMDPs* by [D. Ding et al. 2020]. This approach has been shown to converge globally at an $O(1\sqrt{T})$ rate [Zeng et al. 2022].

While the research community has extensively studied the subjects of safe *RL* and *IRL*, constrained inverse reinforcement learning has received limited attention. [F. Ding and Xue 2022] discusses the enforcement of combinatorial constraints in the *IRL* problem. The question of identifiability of the reward function and generalization to different dynamics and constraints has been studied in [Schlaginhaufen and Kamgarpour 2023]. To the best of our knowledge, there has not been any published research concerning the convergence of an algorithm for solving *CIRL*.

1.2. This thesis

Work	Primal Step	Dual Step	Problem	Regularization	Gradient	Rate (iterations)
Agarwal et al. 2020	PPG	//	RL	no	exact	$O(1/\sqrt{T})$
Agarwal et al. 2020	NPG	//	RL	no	exact	$O(1/T)$
Cen et al. 2021	NPG	//	RL	Shannon	exact	$O(e^{-T})$
Cen et al. 2021	NPG	//	RL	Shannon	perturbed	$O(e^{-T})$
Paternain, Chamon, et al. 2019	Oracle	PGD	CRL	no	exact	$O(1/T)$
Paternain, Calvo-Fullana, et al. 2023	PG	PGD	CRL	no	exact	//
D. Ding et al. 2020	NPG	PGD	CRL	no	exact	$O(1/\sqrt{T})$
D. Ding et al. 2020	NPG	PGD	CRL	no	stochastic	$O(1/\sqrt{T})$
Ziebart, Bagnell, and Dey 2010	DP	GD	IRL	Shannon	exact	//
Zeng et al. 2022	SPI	GD	IRL	Shannon	exact	$O(1/\sqrt{T})$
Chapter 4 of this work	NPG	PGD	CIRL	Shannon	exact	$O(1/\sqrt{T})$
Chapter 5 of this work	NPG	PGD	CIRL	Shannon	stochastic (oracle FIM)	$O(1/\sqrt{T})$

Table 1.1: Comparison of our result with similar works for RL, IRL and CRL (no results exist yet for CIRL). *NPG* denotes natural policy gradients, *PGD* projected gradient descent, *PPG* projected policy gradient, *DP* denotes dynamic programming (requires the dynamics to be known) and *SPI* soft policy iteration (which is described in the background section).

The work presented in this thesis focuses on establishing provable global converge guarantees for an algorithm that solves the *CIRL* problem. The method we introduce, *NPG-CIRL*, implements a primal dual scheme which extends the well-studied natural policy gradient (*NPG*) algorithm and is similar to methods presented to solve the *IRL* [Zeng et al. 2022] and *CRL* [D. Ding et al. 2020]. *NPG-CIRL* differs from the *CRL* algorithm of [D. Ding et al. 2020] by the introduction of entropy regularization and by the fact that two dual variables are studied, with one of them being parametrized

(in this work, we consider linearly parametrized rewards). Our work differs from the *IRL* algorithm of [Zeng et al. 2022] on two main aspects; we consider projected descent steps on reward, whereas they assume that reward is parametrized in some way that doesn't require projection and unlike them, we analyze convergence in the stochastic setting.

The main contribution of this thesis lies in providing a detailed analysis of the *NPG-CIRL* method.

1. We prove that in the exact gradient setting, under softmax policy parameterization, and linear reward parameterization, our method globally converges at a $O(1/\sqrt{T})$ rate.
2. We study the convergence of our algorithm in the stochastic setting, where gradients are estimated using Monte Carlo estimators. We show that the global convergence rate of $O(1/\sqrt{T})$ still holds, even when the gradient estimators are biased.
3. We show that the overall algorithm has a sample complexity of $O(1/\sqrt[4]{S})$. To the best of our knowledge, we provide the first provable global convergence result for an *IRL* algorithm in the stochastic gradient setting.

Finally, we investigate additional structural assumptions on the MDP and the reward and constraint parameterization matrices that lead to a $O(e^{-T})$ convergence rate when satisfied. We have not been able to show that these assumptions are easy to ensure to be true, but these investigations suggest that inquiry in that direction might be interesting.

The results that we present are put into perspective with other state-of-the-art algorithms for similar problems (RL, CRL and IRL) in table 1.1.

2

Background

This chapter introduces the main object we will be studying in this work, Markov decision processes (*MDP*s). In the context of *constrained inverse reinforcement learning*, we will always be dealing with *regularized constrained MDP*s, but for completeness and for greater clarity we will first introduce unregularized *MDP*s (Section 2.1), then regularization (Section 2.2) and finally constraints (Section 2.3).

Once *MDP*s are properly introduced we will discuss the approaches to solving them that we will make use of in this work, *policy gradient methods* (Section 2.4) and *natural policy gradient methods* (Section 2.4.1). Finally, we will introduce and discuss the inverse problem to *MDP*s: *inverse reinforcement learning* (Section 2.5).

2.1. An introduction to Markov Decision Processes (MDPs)

Markov decision processes (*MDP*s) provide a mathematical framework for modelling and studying sequential decision-making in situations with randomness. They have been known and studied since the 1950s originally in the field of Operations Research, notably at the RAND corporation in the United States [Bellman 1957], in parallel they have also appeared in the field of game theory, as a restriction of *Stochastic Games* [Shapley 1953]. Another obvious parallel is that *MDP*s can be thought of as an extension of Markov Chains, which is where they inherit their name from.

We now formally state the definition of *MDP*s as well as the definition of a *Trajectory* generated by an *MDP*. We then illustrate the definition with a simple example. In the following section we will explicitly state that all the results we state are concerning *unregularized MDP*s, the reason we take such care in highlighting that they are not regularized is because we will then completely abandon them in favor of their regularized counterparts (Section 2.2), in our convergence analysis.

Definition 2.1.1 (Unregularized Markov Decision Processes)**.** *A (unregularized) Markov decision process (MDP) is a tuple* $M = (S, A, P, \boldsymbol{r}, \gamma, \boldsymbol{\nu})$ *made up of*

1. *a set of discrete states* $S := \{s_1, s_2, ..., s_n\}$ *(which we call the state-space) of cardinality* $|S| = n$*,*
2. *a set of discrete actions* $A := \{a_1, a_2, ..., a_m\}$ *(which we call the action-space) of cardinality* $|A| = m$*,*
3. *a Markovian transition kernel* $P \in \Delta_S^{S \times A}$ *(which describes the probability* $P(s'|a^{(t)}, s^{(t)})$ *of transitioning to the state* s' *when action* $a^{(t)}$ *is picked while in the state* $s^{(t)}$*),*
4. *a reward function* $\boldsymbol{r} \in \mathcal{R} \subseteq \mathbb{R}^{S \times A}$*,*
5. *a discount factor* $\gamma \in (0, 1]$*,*
6. *an initial state distribution* $\boldsymbol{\nu} \in \Delta_S$*.*

Definition 2.1.2 (Trajectory)**.** *We call a sequence of states and actions generated on an MDP* $M = (S, A, P, \boldsymbol{r}, \gamma, \boldsymbol{\nu})$ *a trajectory, which we denote as:*

$$\tau = \{s^{(0)}, a^{(0)}, s^{(1)}, a^{(2)}, ...\} = \{s^{(i)}, a^{(i}\}_{i=0}^{+\infty}. \tag{2.1}$$

A simple MDP example: the lake cleaning robot In order to clarify how *MDPs* behave and generate trajectories, we will provide a simple example of MDP the *lake cleaning robot* example. In our example we will consider a battery powered robot which has for single goal to clean a lake from waste. In the language of *MDPs* we call our robot, the *agent*. Our *agent* can be in any of 5 states:

State	Situation of the robot
s_1	docked to its charging station,
s_2	offshore with full battery,
s_3	offshore with half-full battery,
s_4	offshore with low battery,
s_5	offshore with an empty battery, lost on the lake.

which together constitute the *state-space* of our *MDP*, we write the state space

$$S = \{s_1, s_2, s_3, s_4, s_5\}. \tag{2.2}$$

At any time step the robot has the possibility to choose one of two actions:

Action	Effect
a_1	navigate in search of waste,
a_2	head to the charging station.

Together actions a_1 and a_2 constitute the action space $A = \{a_1, a_2\}$ of our *MDP*. In an MDP we model time by discrete time-steps, as time progresses, our agent "moves", it changes from one state to another. Furthermore, our agent has a goal (cleaning the lake), in the MDP formalism that goal takes the form of a reward attributed to the agent at each time step depending on the state and action that the agent is in.

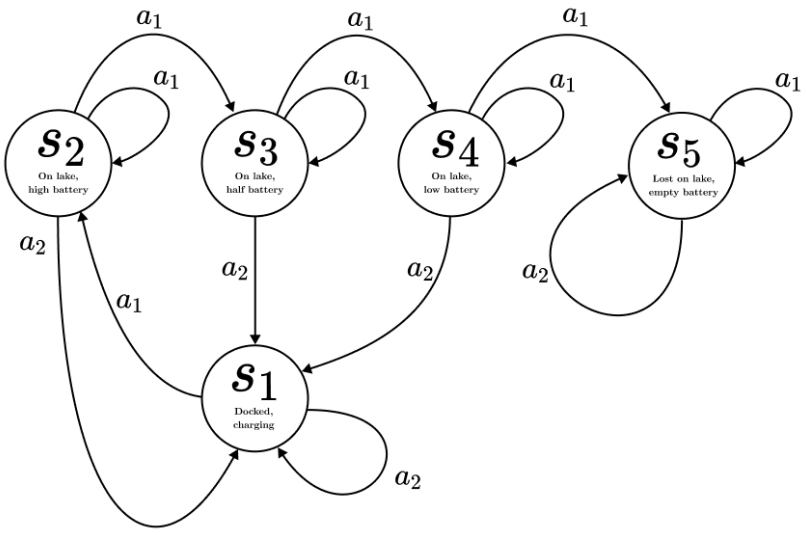


Figure 2.1: A graph representation of possible transitions in the lake cleaning robot *MDP*.

In the case of our lake cleaning robot example, the following transitions are possible, when the robot is docked, the navigate action brings it offshore, on the lake with a fully charged battery. When on the lake the robot has the choice to navigate and thus clean the lake or to come back to its docking state, in which our robot recharges its batteries. Navigating the lake may or may not consume a lot

of energy depending on the environmental conditions. Thus taking the navigate action on the lake lowers the charge of the battery in a stochastic fashion. This is also the goal of our agent, it is when navigating the lake that it cleans it. Therefore, our agent gets a reward of 1 when it picks action a_1 while offshore. It is not known when taking the action whether the battery will lower or not, specifically, there is a probability of 0.5 that taking the navigate action will lower the charge. If the robot ends up in a state where the battery is fully emptied, then all actions have no effects; the robot is lost on the lake and will never be able to charge back up. These possible transitions are pictured in Figure 2.1.

The *MDP*'s Markovian transition kernel P provides a formal way of describing the stochastic evolution from the state of our robot as we move from the time step t to the time step $t+1$. Specifically, it describes how our robot transitions from one state to another by giving the conditional probability $P(s'|s^{(t)}, a^{(t)})$ that the agent transitions to state s' assuming that it currently is in state $s^{(t)}$. For instance, we could write the probability that the robot moves from state s_2 (offshore, full battery) to state s_3 (offshore, half-full battery) when it chooses the action a_1 (navigate) as follows:

$$P(s_3|s_2, a_1) = 0.5. \tag{2.3}$$

The Markovian transition kernel can be thought of as a function $P : S \times A \to \Delta_S$, but since we only care about discrete state and action spaces, i.e. a discrete domain for that function, we can equivalently write it down as a matrix $P \in \mathbb{R}^{nm \times n}$. In our lake-cleaning robot example, the transition kernel P takes the following value:

$$P = \begin{array}{c} \begin{matrix} s_1 & s_2 & s_3 & s_4 & s_5 \end{matrix} \\ \left[\begin{array}{ccccc} 0 & 1 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] \begin{matrix} (s_1, a_1) \\ (s_2, a_1) \\ (s_3, a_1) \\ (s_4, a_1) \\ (s_5, a_1) \\ (s_1, a_2) \\ (s_2, a_2) \\ (s_3, a_2) \\ (s_4, a_2) \\ (s_5, a_2) \end{matrix} \end{array}. \tag{2.4}$$

Here we note that in our formalism, what equation (2.3) refers to is actually just an element of the matrix in equation (2.4), specifically the seventh row, third column of P. This way of indexing elements of a matrix is a bit unusual, but it is arguably the most readable when dealing with *MDP*s, and is the convention that we will follow in this work.

Recall that our agent gets rewarded by a reward of 1, whenever it chooses to navigate, while already on the lake, any other action brings "no reward" i.e. a reward of 0. This brings us to the reward function $\boldsymbol{r} : S \times A \to \mathcal{R}$. Here $\mathcal{R}$ just denotes the set of admissible rewards in our specific *MDP*, in our case the agent only ever gets rewards of 1 or of 0 we have that $\mathcal{R} = \{0, 1\}$. Similarly to what we did for the Markovian transition kernel, we will represent that discrete function as a vector $\boldsymbol{r} \in \mathbb{R}^{nm}$. In our lake cleaning example, the reward vector is the following:

$$\boldsymbol{r} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} (s_1, a_1) \\ (s_2, a_1) \\ (s_3, a_1) \\ (s_4, a_1) \\ (s_5, a_1) \\ (s_1, a_2) \\ (s_2, a_2) \\ (s_3, a_2) \\ (s_4, a_2) \\ (s_5, a_2) \end{matrix}. \tag{2.5}$$

Again, we will write elements of these vector as if they were instances of a reward function, so for example here we could index the third element of the vector as follows:

$$r(s_3, a_1) = 1, \tag{2.6}$$

note that we write the reward vector $\boldsymbol{r}$ in bold and the function $r(s, a)$ in a regular weight, this allows for more clearly differentiating between scalar and vector quantities.

Up until now we have been discussing some key quantities of Definition 2.1.1. We will now conclude this first set of illustrations by considering *trajectories* (Definition 2.1.2). A trajectory, denoted τ is just a sequence generated by the MDP, for instance consider this example of the beginning of a trajectory τ (Figure 2.2). Our agent starts from its docking station in state s_1, and it chooses to pick action a_1 and moves offshore. It then navigates the lake waters and picks actions a_1 three times, the first time it doesn't affect its battery level, but the two next ones it reduces its battery level, it thus reaches low-battery state. At this point it picks action a_2 and sails back to its dock. Note that this is only the beginning of a trajectory since by definition we only consider infinite length trajectories.

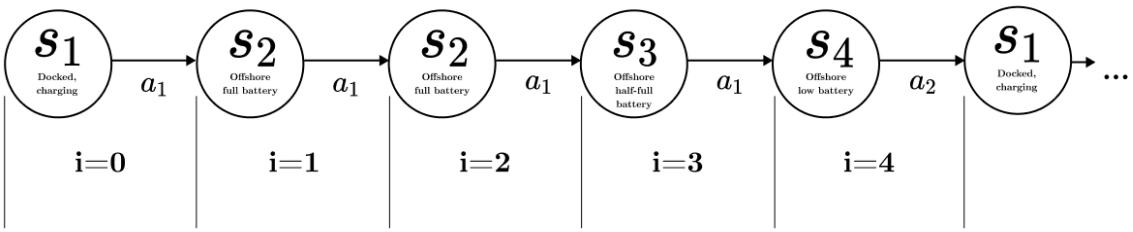


Figure 2.2: Beginning of a trajectory τ over our lake cleaning robot example *MDP*.

The short sequence of state and actions that we described above would be described mathematically in the following notation:

$$\tau = \{s_1, a_1, s_2, a_1, s_2, a_1, s_3, a_1, s_4, a_2, s_1, ...\} \tag{2.7}$$
$$= \{s^{(0)}, a^{(0)}, s^{(1)}, a^{(1)}, s^{(2)}, a^{(2)}, s^{(3)}, a^{(3)}, s^{(4)}, a^{(4)}, s^{(5)}, ...\}. \tag{2.8}$$

In (2.8) we introduce the notation $\square^{(i)}$ which denotes any quantity in the trajectory (reward, state or action) at time step i.

So far we have only discussed how the MDP steps forward with actions picked arbitrarily, we leave our example aside for a moment to formalize the way in which this is done. Policies are function that specify a way of making decisions on some *MDP* we define them more rigorously in the definition below.

Definition 2.1.3 (Policy)**.** *Given a Markov decision process, a policy $\boldsymbol{\pi} \in \Delta_A^S$ is a function that associates an action distribution with each state of an MDP,*

$$\Delta_A^S := \Big\{\boldsymbol{\pi}(\cdot|s) \in \Delta^A,\ \forall s \in S\Big\}, \tag{2.9}$$

where we call Δ_A^S is the policy-set. In plain English, in means the value $\pi(a|s)$ of a policy gives the probability with which an agent using the policy $\boldsymbol{\pi}$ will pick action a when currently in state s.

Implementing a policy in the lake cleaning robot example Back to our example we will now introduce how policies (which are also sometimes referred to as strategies) and we will show how one can implement a simple decision procedure such as Algorithm 1 with a policy function.

Algorithm 1: A simple policy for the lake cleaning robot

At any time step i
switch $s^{(i)}$ **do**
  case $s^{(i)} = s_1$ *(at the dock)* **do**
    pick action a_1 (move to the lake)
  case $s^{(i)} = s_2$ *or* $s^{(i)} = s_3$ *(offshore, battery not low)* **do**
    pick action a_1 (keep navigating)
  case $s^{(i)} = s_4$ *(offshore, low battery)* **do**
    pick action a_2 (head back to dock)
  case $s^{(i)} = s_5$ *(offshore, empty battery)* **do**
    pick between actions a_1 and a_2 uniformly at random

We now consider a policy $\boldsymbol{\pi}$, since the policy is a discrete function, in the same way the reward is, we will also think about is a vector $\boldsymbol{\pi} \in \mathbb{R}^{nm}$, for instance suppose we want our agent to make decisions according to Algorithm 1 we would select the following policy vector:

$$\boldsymbol{\pi}_{\text{clean lake}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0.5 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0.5 \end{bmatrix} \begin{matrix} (s_1, a_1) \\ (s_2, a_1) \\ (s_3, a_1) \\ (s_4, a_1) \\ (s_5, a_1) \\ (s_1, a_2) \\ (s_2, a_2) \\ (s_3, a_2) \\ (s_4, a_2) \\ (s_5, a_2) \end{matrix}. \tag{2.10}$$

We when indexing elements from the policy vector, we use a notation very similar to the one adopted for the reward, suppose we want to index the third element of the policy given in (2.37), we would write it down as:

$$\pi(a_1|s_3)_{\text{clean lake}} = 1, \tag{2.11}$$

where the use of the center bar relates to the probabilistic interpretation of the policy, $\pi(a|s)$ is the conditional probability that the agent picks action a, given that it is in state s .

Here we must highlight a key observation about the behavior of agents in *MDPs* when they are running a fixed policy. The policy fixes the distribution of actions picked by the agent at any given step. This means that when in state s' we can exactly compute the probability that the agent transitions to any other state as follows:

$$P^{\boldsymbol{\pi}}(s'|s) = \sum_{a \in A} \pi(a|s) P(s'|a, s). \tag{2.12}$$

The interpretation of this is quite straightforward: when a policy $\boldsymbol{\pi}$ is fixed for some *MDP* M, the *MDP* becomes a Markov Chain and we can compute its transition kernel from the policy $\boldsymbol{\pi}$ and the transition kernel P of the MDP, as we show in equation (2.12). We call the transition kernel obtained this way the *closed loop transition kernel* and we formally define it in Definition 2.1.4.

Definition 2.1.4 (Closed-loop transition kernel)**.** *Consider an MDP* $M = (S, A, P, \boldsymbol{r}, \gamma, \boldsymbol{\nu})$ *as well as a policy* $\boldsymbol{\pi} \in \Delta_S^A$*, the closed-loop transition kernel* $P^{\boldsymbol{\pi}} \in \Delta_S^S$ *associated with the policy* $\boldsymbol{\pi}$ *gives the probability*

$$P^{\boldsymbol{\pi}}(s'|s) = \sum_{a \in A} \pi(a|s) P(s'|a, s), \tag{2.13}$$

with which the MDP will transition from state s *to state* s'*, assuming that actions are picked according to policy* $\boldsymbol{\pi}$*. It is the transition kernel of the Markov chain created by applying the policy on the MDP.*

Rewards and discounted rewards We will now turn our attention to how the goal of the agent is deduced from the reward. First we introduce the notion of a discounted reward (Definition 2.1.5).

Definition 2.1.5 ((Unregularized) discounted reward)**.** *Consider some MDP $M = (S, A, P, \boldsymbol{r}, \gamma, \boldsymbol{\nu})$, and a trajectory τ we define its (unregularized) discounted reward as follows:*

$$R_{\boldsymbol{r}}(\tau) = (1-\gamma)\sum_{t=0}^{+\infty} \gamma^t r(s^{(t)}, a^{(t)}),\ (s^{(t)}, a^{(t)}) \in \tau. \tag{2.14}$$

We abuse notation and write $(s^{(t)}, a^{(t)}) \in \tau$ to specify that the elements in the sum are the ones given in the trajectory τ.

The discounted reward provides a way to compute something akin to a weighted average of the rewards obtained by the agent which favors immediate rewards compared to rewards far in the future. If we consider the beginning the example trajectory (2.7) that we previously defined, it would receive the following discounted reward:

$$\begin{aligned} R_{\boldsymbol{r}}(\tau) &= R_{\boldsymbol{r}}\big(\{s_1, a_1, s_2, a_1, s_2, a_1, s_3, a_1, s_4, a_2, s_1, ...\}\big) && (2.15)\\ &= (1-\gamma)\big(r(s_1, a_1) + \gamma r(s_2, a_1) + \gamma^2 r(s_2, a_1) + \gamma^2 r(s_3, a_1) + \gamma^2 r(s_4, a_2) + ...\big) && (2.16)\\ &= (1-\gamma)\big(0 + \gamma \cdot 1 + \gamma^2 \cdot 1 + \gamma^2 \cdot 1 + \gamma^2 \cdot 0 + ...\big). && (2.17) \end{aligned}$$

The choice of γ here is key, picking γ close to 1 will increase the importance of rewards further in the future, incentivizing long term planning while a small γ will favor choosing immediate rewards.

Now remember that we are studying stochastic processes, hence the quantity $R_{\boldsymbol{r}}(\tau)$ will change for different trajectories sampled from the *MDP*. We will thus need to study not only the computation of the discounted reward for some trajectory τ but also of the expectation of the quantity $R_{\boldsymbol{r}}(\tau)$. We call that value the *return*.

Definition 2.1.6 ((Unregularized) return)**.** *Given an MDP $M = (S, A, P, \boldsymbol{r}, \gamma, \boldsymbol{\nu})$ as well as a policy $\boldsymbol{\pi} \in \Delta_S^A$, we call (unregularized) return the expected (unregularized) discounted reward under policy $\boldsymbol{\pi}$ (and reward $\boldsymbol{r}$):*

$$J(\boldsymbol{\pi}, \boldsymbol{r}) = \mathbb{E}_{\tau\sim\boldsymbol{\pi}}\Big[R_{\boldsymbol{r}}(\tau)\Big] = (1-\gamma)\mathbb{E}_{\tau\sim\boldsymbol{\pi}}\left[\sum_{t=0}^{+\infty} \gamma^t r(s^{(t)}, a^{(t)})\right] \tag{2.18}$$

The return (Definition 2.1.6) is the quantity that defines the goal of an MPD (Definition 2.1.7)

Definition 2.1.7 (Goal of an MDP and solution map)**.** *Given some reward $\boldsymbol{r} \in \mathcal{R}$, the goal of an MDP is to find policy $\boldsymbol{\pi}_{\boldsymbol{r}}^*$ that maximizes return:*

$$\boldsymbol{\pi}_{\boldsymbol{r}}^* := \mathit{RL}(\boldsymbol{r}) := \arg\max_{\boldsymbol{\pi}\in\Delta_A^S} J(\boldsymbol{\pi}, \boldsymbol{r}). \tag{2.19}$$

We say that a policy $\boldsymbol{\pi}_{\boldsymbol{r}}^*$ solves the MDP for reward $\boldsymbol{r}$. We denote define the solution map to be the mapping $\mathit{RL} : \mathcal{R} \to \Delta_A^S$ that given the reward returns the optimal policy.

Because this work will be focused on *IRL* we write J as a function of both $\boldsymbol{\pi}$, $\boldsymbol{r}$. The notation $\mathbb{E}_{\tau\sim\boldsymbol{\pi}}$ denotes an expectation taken over the probability that the trajectory τ occurs assuming the agent acts according to the policy $\boldsymbol{\pi}$. We can explicitly write out this expectation as:

$$\mathbb{E}_{\tau\sim\boldsymbol{\pi}}\Big[R_{\boldsymbol{r}}(\tau)\Big] = (1-\gamma)\sum_{\tau\in\{\tau\}} P_{\boldsymbol{\pi}}(\tau) R_{\boldsymbol{r}}(\tau) \tag{2.20}$$

$$P_{\boldsymbol{\pi}}(\tau) = \nu(s)\prod_{t=0}^{+\infty} \pi(a^{(t)}, s^{(t)}) P(s^{(t+1)}|s^{(t)}, a^{(t)}),\ (s^{(t)}, a^{(t)}) \in \tau, \tag{2.21}$$

where $\{\tau\}$ denotes the set of all possible trajectories.

We will now introduce a key quantity in the analysis of algorithms running on *MDPs* called the occupancy measure. To see how it takes importance we go back to the equation for return that we just defined:

$$J(\boldsymbol{\pi}, \boldsymbol{r}) = \mathbb{E}_{\tau\sim\boldsymbol{\pi}}\Big[R_{\boldsymbol{r}}(\tau)\Big] = (1-\gamma)\mathbb{E}_{\tau\sim\boldsymbol{\pi}}\left[\sum_{t=0}^{+\infty}\sum_{s,a}\gamma^t r(s^{(t)}, a^{(t)})\right] \tag{2.22}$$

$$\overset{(i)}{=} (1-\gamma)\mathbb{E}_{\tau\sim\boldsymbol{\pi}}\left[\sum_{t=0}^{+\infty}\sum_{s,a}\gamma^t r(s,a)\mathbb{1}(s^{(t)}=s, a^{(t)}=a)\right] \tag{2.23}$$

$$\overset{(ii)}{=} (1-\gamma)\mathbb{E}_{\tau\sim\boldsymbol{\pi}}\left[\sum_{s,a}\sum_{t=0}^{+\infty}\gamma^t r(s,a)\mathbb{1}(s^{(t)}=s, a^{(t)}=a)\right] \tag{2.24}$$

$$= \sum_{s,a} r(s,a)(1-\gamma)\sum_{t=0}^{+\infty}\gamma^t P(s=s^{(t)}, a=a^{(t)}) \tag{2.25}$$

$$= \sum_{s,a} r(s,a)\mu(s,a). \tag{2.26}$$

Where (i) holds by linearity of expectation an (ii) holds by Lebesgue's dominated convergence. What we have done is we have isolated a term $\mu(s,a)$ which measures the "contribution" of a state action pair (s,a) to the return. This measure of "contribution" is what we call the occupancy measure.

Definition 2.1.8 (Occupancy measure)**.** *Given an MDP $M = (S, A, P, \boldsymbol{r}, \gamma, \boldsymbol{\nu})$ and a policy function $\boldsymbol{\pi} \in \Delta_A^S$, we define the occupancy measure $\boldsymbol{\mu^\pi} \in \Delta^{S\times A}$ induced by policy on the MDP as follows, $\forall (s,a) \in S \times A$:*

$$\mu^{\boldsymbol{\pi}}(s,a) = (1-\gamma)\sum_{t=0}^{+\infty}\gamma^t P(s=s^{(t)}, a=a^{(t)}). \tag{2.27}$$

As we previously did for the policy and for the reward, we think of the occupancy measure as a vector $\boldsymbol{\mu} \in \mathbb{R}^{nm}$, and we use the notation $\mu(s,a)$ to index the element of this vector associated with the state action pair (s,a). Here we must underline a key observation, the return is the scalar product of the occupancy measure and the reward vectors, this can easily be seen when writing the equation for the return down:

$$J(\boldsymbol{\pi}, \boldsymbol{r}) = \mathbb{E}_{\tau\sim\boldsymbol{\pi}}\Big[R_{\boldsymbol{r}}(\tau)\Big] = \sum_{s,a}\mu_{\boldsymbol{\pi}}(s,a)r(s,a) = \langle \boldsymbol{\mu_\pi}, \boldsymbol{r}\rangle. \tag{2.28}$$

We also define the state occupancy measure.

Definition 2.1.9 (State occupancy measure)**.** *Given an MDP $M = (S, A, P, \boldsymbol{r}, \gamma, \boldsymbol{\nu})$ and a policy function $\boldsymbol{\pi} \in \Delta_A^S$, we define the state occupancy measure $\boldsymbol{\mu^\pi} \in \Delta^S$ induced by policy on the MDP as follows, $\forall (s,a) \in S \times A$:*

$$\mu_S^{\boldsymbol{\pi}}(s) = (1-\gamma)\sum_{t=0}^{+\infty}\gamma^t P(s=s^{(t)}) = \sum_{a'}\mu^{\boldsymbol{\pi}}(s,a'). \tag{2.29}$$

We call states with state occupancy measure $\mu_S^{\boldsymbol{\pi}}(s) > 0$ *visited*, in the sense that there is a non-zero probability that the agent reaches them in a trajectory. Any state with $\mu_S^{\boldsymbol{\pi}}(s) = 0$ is called *unvisited* and will never be reached by the agent in any trajectory.

We now turn our attention to the study of the properties of the occupancy measure. We first discuss the set of values that can be taken by the occupancy measure. We call this set the occupancy measure set (Definition 2.1.11) and note that although the occupancy measure is contained in the simplex $\Delta^{S\times A}$, the exact set of admissible values it can take must also satisfy what we call the bellman flow

constraints (Definition 2.1.10). Before formally stating the bellman flow constraints, let us introduce the matrix $E \in \mathbb{R}^{nm \times n}$:

$$E = [\overbrace{I_n, ..., I_n}^{m \text{ times}}]^\top. \tag{2.30}$$

Definition 2.1.10 (Bellman flow constraints)**.** *Consider a MDP with Markovian transition kernel $P \in \Delta_S^{S \times A}$, initial state distribution $\boldsymbol{\nu} \in \Delta_S$ and discount factor $\gamma \in [0, 1)$, then any occupancy measure induced by a policy $\boldsymbol{\pi} \in \Delta_A^S$ must satisfy:*

$$(E - \gamma P)^T \boldsymbol{\mu} = (1 - \gamma)\boldsymbol{\nu}.$$

A proof of this result can be found in [Puterman 1994].

The set of feasible occupancy measures is naturally given by Definition 2.1.10.

Definition 2.1.11 (Occupancy measure set)**.** *Given an MDP $M = (S, A, P, \boldsymbol{r}, \gamma, \boldsymbol{\nu})$ and a policy function $\boldsymbol{\pi} \in \Delta_S^A$, we write $\mathcal{M}$ the set of meaningful the occupancy measures (meaningful in the sense that $\exists$ some policy $\boldsymbol{\pi}$ that could result in that specific distribution over the state space, i.e. a distribution satisfying the Bellman flow constraints). The set $\mathcal{M}$ is characterized as:*

$$\mathcal{M} := \left\{ \boldsymbol{\mu} \in \Delta^{S \times A} : (E - \gamma P)^T \boldsymbol{\mu} = (1 - \gamma)\boldsymbol{\nu} \right\}. \tag{2.31}$$

We have already mentioned that the occupancy measure $\boldsymbol{\mu_\pi}$ is induced by a policy. What we mean by that is that for any policy $\boldsymbol{\pi}$ one can compute an occupancy measure $\boldsymbol{\mu}$. Specifically the mapping can be computed from the closed loop transition kernel $P^{\boldsymbol{\pi}}$ and the policy as follows:

$$\boldsymbol{\mu}_S^{\boldsymbol{\pi}} = (1 - \gamma) \sum_{t=0}^{+\infty} (\gamma P^{\boldsymbol{\pi}})^t \boldsymbol{\nu} = (1 - \gamma)(1 - \gamma P^{\boldsymbol{\pi}})^{-1} \boldsymbol{\nu} \tag{2.32}$$

$$\mu_{\boldsymbol{\pi}}(s, a) = \mu_S^{\boldsymbol{\pi}}(s)\pi(a|s). \tag{2.33}$$

This suggests that the mapping from policy to occupancy measure is surjective over the occupancy measure set as defined above. There also exists a one to one mapping in the other direction (altough the mappingi is not injective). Specifically we can recover a policy $\boldsymbol{\pi_\mu}$ that induces the occupancy measure as follows:

$$\pi_{\boldsymbol{\mu}}(a|s) := \begin{cases} \mu(s, a)/\mu_s & \text{if } \mu_s > 0. \\ 1/m & \text{otherwise.} \end{cases} \tag{2.34}$$

Note that the policy picked in an unvisited state has no impact on the occupancy measure, and hence we can just arbitrarily choose any policy for unvisited states. In this work we choose to always associate unvisited states with a policy where we choose actions uniformly at random. This is why we call the mapping almost injective. It is ill-defined for unvisited states, but does not affect our ability, given some occupancy measure $\boldsymbol{\mu}$ to recover a policy $\boldsymbol{\pi_\mu}$ that induces $\boldsymbol{\mu}$. This leads to Proposition 2.1.1.

Proposition 2.1.1 (There exists a one-to-one mapping between occupancy measures and policies)**.** *Any policy $\boldsymbol{\pi} \in \Delta_A^S$ induces an occupancy measure $\boldsymbol{\mu_\pi} \in \mathcal{M}$. Furthermore, we can use equation (2.34) to compute a policy that induces occupancy measure $\boldsymbol{\mu_\pi}$.*

This is discussed more rigorously in [Puterman 1994].

Later on in this work will often make use of this one to one mapping, and we will use the following convention: $\boldsymbol{\pi_\mu}$ is the policy inducing some occupancy measure $\boldsymbol{\mu}$, computed through (2.34), and $\boldsymbol{\mu_\pi}$ is the occupancy measure induced by some policy $\boldsymbol{\pi}$, computed as in (2.32).

Return and occupancy measure in our lake cleaning robot example We come back to our example, to get an intuition we will compute the state occupancy measure for the lake cleaning robot *MDP* under policy $\pi_{\text{clean lake}}$, which we previously defined in (2.37). Assuming that the robot always starts from its dock, i.e. that the initial state distribution is:

$$\boldsymbol{\nu} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} (s_1) \\ (s_2) \\ (s_3). \\ (s_4) \\ (s_5) \end{matrix} \tag{2.35}$$

we can compute the state occupancy measure using equation (2.32):

$$\boldsymbol{\mu}_S^{\boldsymbol{\pi}_{\text{clean lake}}} = (1-\gamma)(I - \gamma P^{\boldsymbol{\pi}_{\text{clean lake}}})^{-1}\boldsymbol{\nu} = \begin{bmatrix} 0.21846 \\ 0.35746 \\ 0.29247 \\ 0.13161 \\ 0 \end{bmatrix} \begin{matrix} (s_1) \\ (s_2) \\ (s_3). \\ (s_4) \\ (s_5) \end{matrix} \tag{2.36}$$

Note that the policy we picked never visits the empty battery state and so state s_5 is unvisited. The occupancy measure can then be computed as in (2.33) an we find:

$$\boldsymbol{\mu}_{\boldsymbol{\pi}_{\text{clean lake}}} = \begin{bmatrix} 0.21846 \\ 0.35746 \\ 0.29247 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.13161 \\ 0 \end{bmatrix} \begin{matrix} (s_1, a_1) \\ (s_2, a_1) \\ (s_3, a_1) \\ (s_4, a_1) \\ (s_5, a_1) \\ (s_1, a_2). \\ (s_2, a_2) \\ (s_3, a_2) \\ (s_4, a_2) \\ (s_5, a_2) \end{matrix} \tag{2.37}$$

It now becomes quite easy to compute the return associated with policy $\boldsymbol{\pi}$ on the lake cleaning robot *MDP*, recall that we can just compute it through the scalar product $\langle \boldsymbol{\mu}_{\boldsymbol{\pi}_{\text{clean lake}}}, \boldsymbol{r} \rangle$, we find:

$$J(\boldsymbol{\pi}, \boldsymbol{r}) = \langle \boldsymbol{\mu}_{\boldsymbol{\pi}_{\text{clean lake}}}, \boldsymbol{r} \rangle = 0.64993. \tag{2.38}$$

Now note that because of Proposition 2.1.1 we know we can equivalently consider policies or occupancy measures and to solve *MDP*s, we also know that the return can be expressed as the scalar product of the occupancy measure $\boldsymbol{\mu}$ and the reward vector $\boldsymbol{r}$. This provides is with a simple way to compute the solution of the MDP (in the sense of Definition 2.1.7). It is easily seen that we can find optimal occupancy measures (and therefore policies) through the following linear program:

$$\max_{\boldsymbol{\mu} \in \mathcal{M}} \langle \boldsymbol{\mu}, \boldsymbol{r} \rangle. \tag{2.39}$$

To complete our overview to *MDP*s, we will introduce two essential quantities which are extremely relevant in the design of algorithms that solve these problems: Q-values and V-values.

Definition 2.1.12 ((Unregularized) value)**.** *Given an MDP $M = (S, A, P, \boldsymbol{r}, \gamma, \boldsymbol{\nu})$ as well as a policy $\boldsymbol{\pi} \in \Delta_S^A$ and some state $s \in S$ we call (unregularized) value the conditional expectation of the (unregularized) discounted reward under policy $\boldsymbol{\pi}$ (and reward $\boldsymbol{r}$), conditioned on the first state in the trajectory being the state s:*

$$V_{\boldsymbol{r}}^{\boldsymbol{\pi}}(s) = \mathbb{E}_{\tau \sim \boldsymbol{\pi}}\Big[R_{\boldsymbol{r}}(\tau)\Big| s_0 = s\Big] = (1-\gamma)\mathbb{E}_{\tau \sim \boldsymbol{\pi}}\Big[\sum_{t=0}^{+\infty} \gamma^t r(s^{(t)}, a^{(t)})\Big| s_0 = s\Big]. \tag{2.40}$$

Definition 2.1.13 ((Unregularized) Q-value)**.** *Given an MDP $M = (S, A, P, \boldsymbol{r}, \gamma, \boldsymbol{\nu})$ as well as a policy $\boldsymbol{\pi} \in \Delta_S^A$ and some state $s \in S$ we call (unregularized) Q-value the conditional expectation of the*

(unregularized) discounted reward under policy π (and reward r), conditioned on the first state in the trajectory being the state s and on the first action picked being a:

$$Q_{\boldsymbol{r}}^{\boldsymbol{\pi}}(s,a) = \mathbb{E}_{\tau\sim\boldsymbol{\pi}}\Big[R_{\boldsymbol{r}}(\tau)\Big|s_0=s,a_0=a\Big] = (1-\gamma)\mathbb{E}_{\tau\sim\boldsymbol{\pi}}\Big[\sum_{t=0}^{+\infty}\gamma^t r(s^{(t)},a^{(t)})\Big|s_0=s,a_0=a\Big]. \tag{2.41}$$

Here we must underline a key relationship between V-value and Q-value: the Q-value can be computed form the V value as follows:

$$Q_{\boldsymbol{r}}^{\boldsymbol{\pi}}(s,a) = (1-\gamma)r(s,a) + \gamma\mathbb{E}_{s'|s,a}\Big[V_{\boldsymbol{r}}^{\boldsymbol{\pi}}(s')\Big]. \tag{2.42}$$

The bellman optimality operator Finally, we introduce a way of solving *MDPs* (in the sense of Definition 2.1.7), the Bellman optimality operator.

Definition 2.1.14 (Bellman optimality operator)**.** *We let $\mathcal{T}:\mathbb{R}^{nm}\to\mathbb{R}^{nm}$ be the operator defined as :*

$$\Big(\mathcal{T}\boldsymbol{Q}_{\boldsymbol{r}}^{\boldsymbol{\pi}}\Big)(s,a) = (1-\gamma)r(s,a) + \gamma\max_{\boldsymbol{\pi}(\cdot|s)\in\Delta_A}\Big(\langle\boldsymbol{\pi}(\cdot|s),\boldsymbol{Q}_{\boldsymbol{r}}^{\boldsymbol{\pi}}(\cdot,a)\rangle\Big), \tag{2.43}$$

where $\big(\langle\boldsymbol{\pi}(\cdot|s),\boldsymbol{V}_{\boldsymbol{r}}^{\boldsymbol{\pi}}(s)\rangle\big)$ denotes a vector of n in which each element associated with state s is computed by the scalar product $\langle\boldsymbol{\pi}(\cdot|s),\boldsymbol{V}_{\boldsymbol{r}}^{\boldsymbol{\pi}}(s)\rangle$.

The bellman optimality operator has two few key properties that make it useful for solving *MDPs*:

1. the vector $\boldsymbol{Q}_{\boldsymbol{r}}^*$ values associated with the optimal policy (in the sense of Definition 2.1.7) is a fixed point of the operator:

$$\mathcal{T}(\boldsymbol{Q}_{\boldsymbol{r}}^*) = \boldsymbol{Q}_{\boldsymbol{r}}^*, \tag{2.44}$$

2. the operator is a γ-contraction in the $\|\cdot\|_\infty$ norm. For any two Q-value vectors $\boldsymbol{Q}$ and $\bar{\boldsymbol{Q}}$, we have:

$$\|\mathcal{T}\boldsymbol{Q} - \mathcal{T}\bar{\boldsymbol{Q}}\|_\infty \leq \gamma\|\boldsymbol{Q}-\bar{\boldsymbol{Q}}\|_\infty. \tag{2.45}$$

These two properties (which are discussed and analyzed in details in [Puterman 1994]) allow, in *MDPs* with known dynamics, to use the bellman optimality operator for solving the *MDPs*. Iteratively running the operator $\mathcal{T}$ on some arbitrarily initial $\boldsymbol{Q}$ converges at a linear rate to the optimal $\boldsymbol{Q}$-values $\boldsymbol{Q}^*$. This method is known as *value iteration*.

2.2. Regularization in MDPs

We now consider regularized *MDPs*, an extension of *MDPs* in which we introduce a convex regularizer function $\Omega:\Delta_A\to\mathbb{R}$. The topic of regularized *MDPs* is of interest to us because (as we will later discuss) regularization ensures that the optimal solution to an *MDP* is unique. Regualrization is also required to make the *IRL* problem non-degenerate.

What differentiates a regularized *MDP* from an unregularized one is the introduction of a convex regularization function Ω which is weighted by regularization factor β. The regularizer affects the computation of the return, of the Q-value and of the V-value function and thus changes what constitutes an optimal solution to the *MDP*. On the other hand the introduction of regularizer does not affect the dynamics of the *MDP*, so for instance the occupancy measure set $\mathcal{M}$ is exactly the same whether the *MDP* is regularized or not.

Return, Q and V-values in regularized MDPs The regularization is applied to the discounted reward and thus affects the return, Q and V-values as well as the optimal policies. We now define the discounted reward and return, in the context of regularized *MDPs*.

Definition 2.2.1 (Discounted reward (regularized))**.** *Given a trajectory τ, generated by some policy $\boldsymbol{\pi} \in \Delta_S^A$ on a regularized MDP $M = (S, A, P, \boldsymbol{r}, \gamma, \boldsymbol{\nu}, \Omega, \beta)$, we define its (regularized) discounted reward as follows:*

$$R_{\boldsymbol{r}}(\tau) = (1-\gamma)\sum_{t=0}^{+\infty}\gamma^t(r(s^{(t)}, a^{(t)}) - \beta\Omega(\boldsymbol{\pi}(\cdot|s_t))),\ (s^{(t)}, a^{(t)}) \sim \tau. \tag{2.46}$$

This way of regularizing the MDP penalizes policies that are "too deterministic" and can thus be though of as a way to incentivize exploration.

Definition 2.2.2 (Return (regularized))**.** *Given an MDP $M = (S, A, P, \boldsymbol{r}, \gamma, \boldsymbol{\nu}, \Omega, \beta)$ as well as a policy $\boldsymbol{\pi} \in \Delta_S^A$, we call (regularized) return the expected discounted reward under policy $\boldsymbol{\pi}$ (and reward $\boldsymbol{r}$):*

$$J(\boldsymbol{\pi}, \boldsymbol{r}) = \mathbb{E}_{\tau\sim\boldsymbol{\pi}}\Big[R_{\boldsymbol{r}}(\tau)\Big] = (1-\gamma)\mathbb{E}_{\tau\sim\boldsymbol{\pi}}\Big[\sum_{t=0}^{+\infty}\gamma^t(r(s^{(t)}, a^{(t)}) - \beta\Omega(\boldsymbol{\pi}(\cdot|s_t))\Big] \tag{2.47}$$

$$= \mathbb{E}_{(s,a)\sim\boldsymbol{\mu}^{\boldsymbol{\pi}}}\Big[r(s^{(t)}, a^{(t)})\Big] - \beta\mathbb{E}_{(s,a)\sim\boldsymbol{\mu}^{\boldsymbol{\pi}}}\Big[\Omega(\boldsymbol{\pi}(\cdot|s_t))\Big] \tag{2.48}$$

In equation 2.48 the linearity of expectation, we are able to isolate:

$$\mathbb{E}_{(s,a)\sim\boldsymbol{\mu}^{\boldsymbol{\pi}}}\Big[\Omega(\boldsymbol{\pi}(\cdot|s_t))\Big], \tag{2.49}$$

this quantity will appear in the computation of all reward-related quantities, we will thus name it *expected regularizer* (Definition 2.2.3) and we will characterize some of its properties.

Definition 2.2.3 (Expected regularizer)**.** *Given an MDP $M = (S, A, P, \boldsymbol{r}, \gamma, \boldsymbol{\nu}, \Omega, \beta)$ as well as an occupancy measure $\boldsymbol{\mu}$, we let the function $\tilde{\Omega} : \boldsymbol{\mu}^{S\times A}$ denote the expectation:*

$$\tilde{\Omega}(\boldsymbol{\mu}) = \mathbb{E}_{s\sim\boldsymbol{\mu}_S}\Big[\Omega(\boldsymbol{\pi}(\cdot|s))\Big] = (1-\gamma)\mathbb{E}_{\tau\sim\pi}\Big[\sum_{t=0}^{+\infty}\gamma^t\Omega(\boldsymbol{\pi}(\cdot|s_t))\Big]. \tag{2.50}$$

of its regularizer under occupancy measure and policy $\boldsymbol{\pi}$.

Proposition 2.2.1 (Strict convexity is preserved in the expected regularizer)**.** *Assuming that the regularizer function $\Omega : \Delta_A \to \mathbb{R}$ is strictly convex, the expected regularizer also is strictly convex.*

This result is proven in [Schlaginhaufen and Kamgarpour 2023].

With the expected regularizer defined, we can express the return function in the more concise, scalar product form:

$$J(\boldsymbol{\pi}, \boldsymbol{r}) = \mathbb{E}_{(s,a)\sim\boldsymbol{\mu}^{\boldsymbol{\pi}}}\Big[r(s^{(t)}, a^{(t)})\Big] - \beta\mathbb{E}_{(s,a)\sim\boldsymbol{\mu}^{\boldsymbol{\pi}}}\Big[\Omega(\boldsymbol{\pi}(\cdot|s_t))\Big] = \langle \boldsymbol{r}, \boldsymbol{\mu}^{\boldsymbol{\pi}}\rangle - \tilde{\Omega}(\boldsymbol{\mu}). \tag{2.51}$$

We now formally state the definition of the Q and V-values in the regularized setting.

Definition 2.2.4 (V-value (regularized))**.** *Given an MDP $M = (S, A, P, \boldsymbol{r}, \gamma, \boldsymbol{\nu}, \Omega, \beta)$ as well as a policy $\boldsymbol{\pi} \in \Delta_S^A$ and some state $s \in S$ we call (regularized) V-value the conditional expectation of the (regularized) discounted reward under policy $\boldsymbol{\pi}$ (and reward $\boldsymbol{r}$), conditioned on the first state in the trajectory being the state s:*

$$V_{\boldsymbol{r}}^{\boldsymbol{\pi}}(s) = \mathbb{E}_{\tau\sim\boldsymbol{\pi}}\Big[R_{\boldsymbol{r}}(\tau)\Big|s_0 = s\Big] = (1-\gamma)\mathbb{E}_{\tau\sim\boldsymbol{\pi}}\Big[\sum_{t=0}^{+\infty}\gamma^t r(s^{(t)}, a^{(t)}) - \beta\Omega(\boldsymbol{\pi}(\cdot|s_t))\Big|s_0 = s\Big]. \tag{2.52}$$

Definition 2.2.5 (Q-value (regularized))**.** *Given an MDP $M = (S, A, P, \boldsymbol{r}, \gamma, \boldsymbol{\nu}, \Omega, \beta)$ as well as a policy $\boldsymbol{\pi} \in \Delta_S^A$ and some state $s \in S$ we call (regularized) Q-value the conditional expectation of the (regularized) discounted reward under policy $\boldsymbol{\pi}$ (and reward $\boldsymbol{r}$), conditioned on the first state in the trajectory being the state s and on the first action picked being a:*

$$Q_{\boldsymbol{r}}^{\boldsymbol{\pi}}(s, a) = \mathbb{E}_{\tau\sim\boldsymbol{\pi}}\Big[R_{\boldsymbol{r}}(\tau)\Big|s_0 = s, a_0 = a\Big] \tag{2.53}$$

$$= (1-\gamma)\mathbb{E}_{\tau\sim\boldsymbol{\pi}}\Big[\sum_{t=0}^{+\infty}\gamma^t r(s^{(t)}, a^{(t)}) - -\beta\Omega(\boldsymbol{\pi}(\cdot|s_t))\Big|s_0 = s, a_0 = a\Big]. \tag{2.54}$$

The soft bellman optimality operator We will now discuss the counterpart to the bellman optimality operator in the regularized setting: the *soft bellman optimality operator*. Recall that in the unregularized setting, we defined the *bellman optimality operator* as:

$$\Big(\mathcal{T}\boldsymbol{Q}_{\boldsymbol{r}}^{\boldsymbol{\pi}}\Big)(s,a) = (1-\gamma)r(s,a) + \gamma \max_{\boldsymbol{\pi}(\cdot|s)\in\Delta_A} \Big(\langle \boldsymbol{\pi}(\cdot|s), \boldsymbol{Q}_{\boldsymbol{r}}^{\boldsymbol{\pi}}(\cdot,a)\rangle\Big). \tag{2.55}$$

When working within the regularized MDP setting, our operator takes the form of the soft bellman optimality operator.

Definition 2.2.6 (Soft-bellman optimality operator)**.** *We let $\mathcal{T}_\beta : \mathbb{R}^{nm} \to \mathbb{R}^{nm}$ be the operator defined as :*

$$\Big(\mathcal{T}_\beta\boldsymbol{Q}_{\boldsymbol{r}}^{\boldsymbol{\pi}}\Big)(s,a) = (1-\gamma)r(s,a) + \gamma \max_{\boldsymbol{\pi}(\cdot|s)\in\Delta_A} \Big(\langle \boldsymbol{\pi}(\cdot|s), \boldsymbol{Q}_{\boldsymbol{r}}^{\boldsymbol{\pi}}(\cdot,a)\rangle - \beta\tilde{\Omega}(\boldsymbol{\pi}(\cdot|s))\Big). \tag{2.56}$$

Similarly to its unregularized counterpart, the soft-bellman optimality operator satisfies properties that make it an efficient tool for solving *MDP*s. We formalize these properties in Proposition 2.2.2.

Proposition 2.2.2 (Properties of the soft bellman optimality operator)**.** *The operator $\mathcal{T}_\beta : \mathbb{R}^{nm} \to \mathbb{R}^{nm}$ as defined in Definition 2.2.6, satisfies the following properties:*

1. *the vector $\boldsymbol{Q}_{\boldsymbol{r}}^*$ values associated with the optimal policy (in the sense of Definition 2.1.7) is a fixed point of the operator:*

$$\mathcal{T}(\boldsymbol{Q}_{\boldsymbol{r}}^*) = \boldsymbol{Q}_{\boldsymbol{r}}^*, \tag{2.57}$$

2. *the operator is a γ-contraction in the $\|\cdot\|_\infty$ norm. For any two Q-value vectors $\boldsymbol{Q}$ and $\bar{\boldsymbol{Q}}$, we have:*

$$\|\mathcal{T}\boldsymbol{Q} - \mathcal{T}\bar{\boldsymbol{Q}}\|_\infty \leq \gamma\|\boldsymbol{Q} - \bar{\boldsymbol{Q}}\|_\infty. \tag{2.58}$$

These properties are proven in [Geist, Scherrer, and Pietquin 2019].

2.2.1. Entropy regularized MDPs

Next, we discuss one very specific example of regularizer $\Omega : \Delta^A \to \mathbb{R}$ that is particularly well-studied is that of the negative Shannon Entropy. Which we first formally define.

Definition 2.2.7 (Shannon Entropy)**.** *Consider some distribution $\boldsymbol{p} \in \Delta^X$ over the discrete random variable X, we define the Shannon entropy $H : \Delta^X \to \mathbb{R}$ as follows:*

$$H(X) = -\sum_{i=0}^{|X|} p_i \log(p_i). \tag{2.59}$$

One key property we will make use of is that the soft bellman operator can be computed in closed form when using the negative Shannon entropy as a regularizer.

Proposition 2.2.3 (Closed-form of the soft bellman optimality operator when $\Omega = -H$)**.** *When considering a Shannon-regularized MDP, the operator $\mathcal{T}_\beta : \mathbb{R}^{nm} \to \mathbb{R}^{nm}$ admits the following closed-form solution:*

$$\mathcal{T}(\boldsymbol{Q}_{\boldsymbol{r}}^*) = (1-\gamma)r(s,a) + \gamma\mathbb{R}_{s'|s,a}\Big[\beta\|\exp \boldsymbol{Q}_{\boldsymbol{r}}(s'|\cdot)\|_1\Big]. \tag{2.60}$$

This result is proved in [Cen et al. 2021].

When using the Shannon entropy as a regularizer the following relationships between V-value and Q-value hold:

$$Q_{\boldsymbol{r}}^{\boldsymbol{\pi}}(s,a) = (1-\gamma)r(s,a) + \gamma\mathbb{E}_{s'|s,a}[V(s')], \tag{2.61}$$

$$V_{\boldsymbol{r}}^{\boldsymbol{\pi}}(s) = \mathbb{E}_{a\sim\boldsymbol{\pi}}[Q_{\boldsymbol{r}}^{\boldsymbol{\pi}}(s,a) - \beta\log\pi(a|s)]. \tag{2.62}$$

These equalities were first derived in [Nachum et al. 2017].

And we also have a direct way of computing the optimal policy $\boldsymbol{\pi}_{\boldsymbol{r}}^*$ from the optimal Q-values, we formalize it in the proposition below.

Proposition 2.2.4 (Optimal policy is softmax of the optimal Q-values)**.** *When considering a Shannon entropy regularized MDP the optimal policy is directly computable as a function of the Q-values as follows:*

$$\pi^*_{\boldsymbol{r}}(s,a) = \frac{\exp(Q)^*_{\boldsymbol{r}}(s,a)}{\sum_{a\in A}\exp(Q)^*_{\boldsymbol{r}}(s,a)}. \tag{2.63}$$

This is proved in [Nachum et al. 2017].

2.3. Constraints in MDPs : CMDPs

Next, we introduce how constraints can be introduced in Markov decision processes. We define the *Constrained Markov Decision Process* (*CMPD*) setting (Definition 2.3.1).

Definition 2.3.1 (Constrained Markov Decision Process)**.** *A Constrained Markov decision process (CMDP) is a tuple $M = (S, A, P, \boldsymbol{r}, \gamma, \boldsymbol{\nu}, \Omega, \beta)$ made up of*

1. *a set of discrete states $S := \{s_1, s_2, ..., s_n\}$ (which we call the state-space) of cardinality $|S| = n$,*
2. *a set of discrete actions $A := \{a_1, a_2, ..., a_m\}$ (which we call the action-space) of cardinality $|A| = m$,*
3. *a Markovian transition kernel $P \in \Delta_S^{S\times A}$ (which describes the probability $P(s'|a, s)$ of transitioning to the state s' when action a is picked while in the state s),*
4. *a reward function $\boldsymbol{r} \in \mathcal{R} \subseteq \mathbb{R}^{S\times A}$,*
5. *a discount factor $\gamma \in (0, 1]$,*
6. *an initial state distribution $\boldsymbol{\nu} \in \Delta_S$,*
7. *a cost matrix $\Psi \in \mathbb{R}^{nm\times d}$,*
8. *a constraint vector $\boldsymbol{b} \in \mathbb{R}^d$,*
9. *a convex regularizer function $\Omega : \Delta_A \to \mathbb{R}$,*
10. *a regularization parameter $\beta \in \mathbb{R}^+$.*

The distinction between an *MDP* and a *CMPD* lies in the introduction of a set of d constraints, represented by a vector $\boldsymbol{b} \in \mathbb{R}^d$ and of d cost functions that we represent through a cost matrix $\Psi \in \mathbb{R}^{nm\times d}$. The constraints are considered satisfied when all costs are lower than the constraints, i.e. when:

$$\Psi^\top \boldsymbol{\mu} \leq \boldsymbol{b}. \tag{2.64}$$

The constraint matrix Ψ can be though of a block matrix made up d cost columns $[\boldsymbol{\psi}_1, ...\boldsymbol{\psi}_d]$. Once written out in this form it is easy to see that the cost expression $\Psi^\top \boldsymbol{\mu}$ is equivalent to the computation multiple return (Definition 2.1.6) functions:

$$\Psi^\top \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\psi}_1 \\ \vdots \\ \boldsymbol{\psi}_d \end{bmatrix} \boldsymbol{\mu} = \begin{bmatrix} \langle \boldsymbol{\psi}_1, \boldsymbol{\mu}\rangle \\ \vdots \\ \langle \boldsymbol{\psi}_d, \boldsymbol{\mu}\rangle \end{bmatrix} \leq \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}. \tag{2.65}$$

We have chosen to introduce the cost as a matrix-vector product of matrix Ψ and the occupancy measure $\boldsymbol{\mu}$. But the cost matrix defines a cost vector $\boldsymbol{\Psi})(s, a)$ for each state-action pair $(s, a) \in S \times A$ and we can alternatively think of the cost vectors in this way. The two formulations are equivalent:

$$\Psi^\top \boldsymbol{\mu} = \begin{bmatrix} \langle \boldsymbol{\psi}_1, \boldsymbol{\mu}\rangle \\ \vdots \\ \langle \boldsymbol{\psi}_d, \boldsymbol{\mu}\rangle \end{bmatrix} = \begin{bmatrix} (1-\gamma)\mathbb{E}_{\tau\sim\boldsymbol{\pi}}\left[\sum_{i=0}^{\infty}\gamma^i \boldsymbol{\psi}_1(s^{(i)}, a^{(i)})\right] \\ \vdots \\ (1-\gamma)\mathbb{E}_{\tau\sim\boldsymbol{\pi}}\left[\sum_{i=0}^{\infty}\gamma^i \boldsymbol{\psi}_d(s^{(i)}, a^{(i)})\right] \end{bmatrix} = (1-\gamma)\mathbb{E}_{\tau\sim\boldsymbol{\pi}}\left[\sum_{i=0}^{\infty}\gamma^i \boldsymbol{\Psi}(s^{(i)}, a^{(i)})\right]. \tag{2.66}$$

Recall that for unconstrained MDPs we defined the solution map (Definition 2.1.7) as:

$$\boldsymbol{\pi}^*_{\boldsymbol{r}} := \text{RL}(\boldsymbol{r}) := \arg\max_{\boldsymbol{\pi}\in\Delta_A^S} J(\boldsymbol{\pi}, \boldsymbol{r}), \tag{2.67}$$

in the CMPD setting we will re-define our solution map to that enforce that the constraints are satisfied.

Definition 2.3.2 (CRL solution map)**.** *In constrained Markov decision processes (CMDPs) we define the solution map to be the mapping CRL* $: \mathcal{R} \to \Delta_A^S$ *that given the reward returns the optimal policy. That mapping is given by:*

$$\begin{aligned} \textit{CRL}(\boldsymbol{r}) := \arg\max_{\boldsymbol{\pi} \in \Delta_A^S} & J(\boldsymbol{\pi}, \boldsymbol{r}), \\ \textit{s.t. } & \Psi^\top \boldsymbol{\mu}_{\boldsymbol{\pi}} \leq \boldsymbol{b}. \end{aligned} \tag{2.68}$$

We are now done characterizing the types of *MDP*s that we will consider in this work. We will move on to the description of possible methods to solve them.

2.4. Policy Gradient methods

From an optimization standpoint, the problem of finding the solution to *MDP*s can be thought of as an optimization problem that solves:

$$\arg\max_{\boldsymbol{\pi} \in \Delta_A^S} J(\boldsymbol{\pi}, \boldsymbol{r}). \tag{2.69}$$

Policies gradient methods approach this problem by running gradient ascent schemes on the return function J either directly on policy vectors $\boldsymbol{\pi}$ [Ronald J. Williams 1988] or, when the policy is parameterized, by some parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{nm}$:

$$\boldsymbol{\pi}^{(t+1)} \leftarrow \boldsymbol{\pi}^{(t)} + \eta_\theta \nabla_{\boldsymbol{\pi}} J(\boldsymbol{\pi}, \boldsymbol{r}), \qquad \boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} + \eta_\theta \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}}, \boldsymbol{r}). \tag{2.70}$$

We denote the policy parameterized by the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{nm}$ as $\boldsymbol{\pi}_{\boldsymbol{\theta}}$. The most common policy parameterization used in for RL is the tabular softmax parameterization, which is simply given by:

$$\pi_{\boldsymbol{\theta}}(a|s) = \frac{\exp\big(\theta(s,a)\big)}{\sum_{a' \in A} \exp\big(\theta(s,a')\big)}. \tag{2.71}$$

There are multiple algorithms that work with the general idea of policy gradient, but in the context of this work, we will focus on the natural policy gradient (*NPG*) algorithm.

2.4.1. Natural Policy Gradients

NPG was first introduced in [Kakade 2001] and thoroughly analyzed for the entropy regularized setting[1] in [Cen et al. 2021] and [Mei et al. 2020]. In the following section, we introduce the algorithm and discuss a few key results that we will require for the analysis of our own algorithm. The main insight for the original *NPG* paper [Kakade 2001] is to precondition the gradient steps with the moore penrose inverse of the Fisher information matrix of the policy parametrization.

Definition 2.4.1 (NPG policy step)**.** *The natural policy gradient step is given by:*

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} + \eta_\theta \big(\mathfrak{F}^{\boldsymbol{\theta}}\big)^\dagger \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \boldsymbol{r}) \tag{2.72}$$

The idea behind the preconditioning is to ensure that gradient direction is taken on the manifold that the policy parameter $\boldsymbol{\theta}$ parameterizes. This hinges on the fact that the Fisher information matrix is up to scale, an invariant metric on the space of parameterized probability (in our case policy) distributions.

Definition 2.4.2 (The Fisher information matrix)**.** *The Fisher information matrix (FIM) is given by:*

$$\big[\mathfrak{F}^\theta\big](s,a) = \mathbb{E}_{(s,a)\sim\boldsymbol{\mu}^\theta}\Big[\nabla_{\boldsymbol{\theta}} \log \pi(a|s) (\nabla_{\boldsymbol{\theta}} \log \pi(a|s))^\top\Big]. \tag{2.73}$$

One of the results concerning *NPG* is that when the algorithm is run on a softmax parameterized policy in an entropy-regularized setting, the updates in the policy take an exponentiated gradient form.

[1] Note that for the setting without entropy regularization, an analysis was developed in [Agarwal et al. 2020].

Lemma 2.4.1 (The softmax NPG step with entropy regularization yields MWU in the policy)**.** *Consider the NPG policy step (as specified in Definition 2.4.1) applied on a tabular softmax parameterized policy in a Shannon entropy regularized MDP. Assuming that* $0 < \eta_\theta \leq 1/\beta$ *we have that the policy updates take the form, for any* $(s, a) \in S \times A$*:*

$$\pi_{\boldsymbol{\theta}^{(t+1)}}(a|s)^{(t+1)} \leftarrow \frac{1}{Z^{(t)}(s)}(\pi_{\boldsymbol{\theta}^{(t)}}(a|s)^{(t)})^{1-\eta_\theta\beta} \exp\big(\eta_\theta Q_{\boldsymbol{r}}^{\boldsymbol{\theta}^{(t)}}(s, a)\big), \tag{2.74}$$

where $Z^{(s)}(s)$ *ensures that* $\sum_a \pi^{(t+1)}(a|s) = 1$*, this update rule resembles multiplicative weight update (abbreviated MWU).*

This result is proved in [Cen et al. 2021].

When the learning rate $\eta_\theta = 1/\beta$ is picked, it is easily seen that update of equation (2.75) reduces to a simpler form, which is independent of the previous policy $\boldsymbol{\pi}_{\boldsymbol{\theta}^{(t)}}$. We call this special instance of *NPG* soft policy iteration.

Proposition 2.4.1 (Soft Policy Iteration)**.** *We consider the NPG step in the entropy-regularized setting with tabular softmax parameterized policies. When the special learning rate* $\eta_\theta = \frac{1}{\beta}$ *is picked, then the policy-updates take the form:*

$$\pi_{\boldsymbol{\theta}^{(t+1)}}(a|s)^{(t+1)} \leftarrow \frac{1}{Z^{(t)}(s)} \exp\big(\eta_\theta Q_{\boldsymbol{r}}^{\boldsymbol{\theta}^{(t)}}(s, a)\big), \tag{2.75}$$

where $Z^{(s)}(s)$ *ensures that* $\sum_a pi^{(t+1)}(a|s) = 1$*, we call iterations of this form soft policy iterations.*

The reason we give special care to the soft policy iteration case is because the soft iterates implement the bellman optimality operator (Definition 2.2.6).

Proposition 2.4.2 (Soft Policy Iteration implements the soft Bellman optimality operator)**.** *We consider the* Q*-values* $\boldsymbol{Q}_{\boldsymbol{r}}^{(t)}$ *and* $\boldsymbol{Q}_{\boldsymbol{r}}^{(t+1)}$ *associated with policies* $\boldsymbol{\pi}_{\boldsymbol{\theta}^{(t)}}$ *and* $\boldsymbol{\pi}_{\boldsymbol{\theta}^{(t+1)}}$ *generated through one step of soft policy iteration. It holds that:*

$$\mathcal{T}(\boldsymbol{Q}_{\boldsymbol{r}}^{(t)}) = \boldsymbol{Q}_{\boldsymbol{r}}^{(t+1)} \tag{2.76}$$

So running soft policy iterations is equivalent to using the soft bellman optimality operator on the Q*-values.*
This result is proved in [Cen et al. 2021].

Proposition 2.4.1 allows for easily verifying that in the soft policy iteration setting, NPG converges linearly fast. This is the main driver behind most of the analysis developed in [Cen et al. 2021], and will be a workhorse of our own analysis. Another key result that we will make use of in some of our derivations it that NPG, regardless of the learning rates, ensures monotonous improvement in the Q-values.

Lemma 2.4.2 (Monotonous improvement of exact NPG)**.** *Consider the NPG policy step (as specified in Definition 2.4.1) applied on a tabular softmax parameterized policy, in a Shannon entropy regularized MDP. Assuming that* $0 < \eta_\theta \leq 1/\beta$ *we have that for any* $(s, a) \in S \times A$*, the following inequality holds:*

$$Q_{\boldsymbol{r}}^{(t+1)}(s, a) \geq Q_{\boldsymbol{r}}^{(t)}(s, a). \tag{2.77}$$

This result is proved in [Cen et al. 2021].

The last result concerning *NPG* that we will make use of in our analysis is the so-called soft suboptimality lemma.

Lemma 2.4.3 (Soft suboptimality)**.** *Consider an entropy-regularized MPC and let* $\boldsymbol{\pi}^* \in \Delta_A^S$ *be an optimal policy on that MDP, then the following bound on the suboptimality of any other policy holds:*

$$J(\boldsymbol{\pi}^*, \boldsymbol{r}) - J(\boldsymbol{\pi}, \boldsymbol{r}) = \beta \mathbb{E}_{s\sim\boldsymbol{\mu}_S^{\boldsymbol{\pi}}}\Big[D_{KL}\big(\boldsymbol{\pi}(\cdot|s)||\boldsymbol{\pi}^*(\cdot|s)\big)\Big]. \tag{2.78}$$

2.4.2. Extending Policy Gradient methods to Constrained RL

The objective of a *CMDP* can be thought of as an optimization problem of the form:

$$\arg\max_{\boldsymbol{\pi}\in\Delta_A^S} J(\boldsymbol{\pi}, \boldsymbol{r}), \tag{2.79}$$

$$\text{s.t. } \Psi^\top \boldsymbol{\mu}_{\boldsymbol{\pi}} \leq \boldsymbol{b}. \tag{2.80}$$

Solving Constrained Reinforcement Learning (*CRL*) problems is performed with no direct access to the *CMDP* dynamics, hence constraint satisfaction cannot be enforced through projection methods. For that reason, the most common approach has been to use the method of lagrangian multipliers on the *CRL* problem. The *CRL* lagrangian takes the form [Paternain, Chamon, et al. 2019]:

$$\arg\max_{\boldsymbol{\pi}\in\Delta_A^S}\min_{\boldsymbol{\lambda}\in\Lambda} J(\boldsymbol{\pi}, \boldsymbol{r}) + \langle \boldsymbol{\lambda}, \boldsymbol{b} - \Psi^\top \boldsymbol{\mu}^{\boldsymbol{\pi}_\theta}\rangle. \tag{2.81}$$

This marks a fundamental difference between RL and CRL methods: while RL problems are maximization problems, CRL problems are saddle point problems. Assuming access to Oracle solution for the policy optimization side of the min-max problem, [Paternain, Chamon, et al. 2019] have shown that the problem (2.81) reduces to gradient descent on the dual and by dual convexity, global convergence. Furthermore, [Paternain, Calvo-Fullana, et al. 2023] have proposed using policy gradient methods in the primal, as a way of approximating the primal solution Oracle, but have not shown global convergence. Global convergence for a practical algorithm has been shown by [D. Ding et al. 2020] who proposes the use Natural Policy Gradient (without entropy regularization) in the primal and show a global convergence rate of $O(1/\sqrt{T})$.

2.5. Inverse Reinforcement Learning

To conclude this background chapter, we introduce the inverse problem in *MDPs*, *inverse reinforcement learning (IRL)*. The *IRL* problem addresses the question of recovering the reward given either directly an expert policy or some dataset providing information about this policy. We define *IRL* as the search of a right-inverse solution map to the *RL* solution map (Definition 2.1.7).

Definition 2.5.1 (CIRL solution map)**.** *The exact solution map of the IRL problem is a mapping IRL* : $\Delta_A^S \to \mathcal{R}$ *which satisfies:*

$$(\textit{RL} \circ \textit{IRL})(\boldsymbol{\pi}^E) = \boldsymbol{\pi}^E. \tag{2.82}$$

A key observation to be made about the *IRL* problem is that is ill-defined. Many choices of reward can lead to the same optimal policy. One trivial example is that any policy π is optimal with respect to constant rewards.

Different approaches have been proposed to overcome this challenge in IRL, one of the most promising one is that of maximum causal entropy IRL (*MCE-IRL*), first proposed by [Ziebart, Bagnell, and Dey 2010], which ensures the uniqueness of the solution to the IRL problem by introducing entropy regularization.

3

Problem Setting

We now circle back to the main problem at hand, in the following chapter we formally introduce the *Constrained Inverse Reinforcement Learning Problem* (*CIRL*), discuss how it can be reduced to an optimization problem and what quantities we will use to measure the quality of approximate solutions.

3.1. Constrained Inverse Reinforcement Learning

The *CIRL* problem is the extension of the *IRL* problem to *CMDPS*. *IRL*, which we previously introduced in Section 2.5, is the problem of, given expert data, inferring a reward for which the expert that produced the data is optimal. In *CMDPs*, we extend *MDPs* with constraints, as discussed in Section 2.3. The *CIRL* problem can be defined in two ways, either in an idealized fashion, where direct access to the expert policy $\boldsymbol{\pi}^E$ is assumed. Alternatively, we can define *CIRL* as a problem where the expert policy $\boldsymbol{\pi}^E$ is only accessible through a dataset of example trajectories $\mathcal{D}$. We first define the simplest direct policy access setting.

Problem Definition 3.1.1 (CIRL with direct expert policy access)**.** *In the direct expert policies access CIRL problem, we have access to: a regularized Constrained Markov Decision Processes (CMDP) for which we do not know the reward*

$$\textit{CMDP} \setminus \boldsymbol{r} = (S, A, P, \boldsymbol{\nu}, \Psi, \boldsymbol{b}, \gamma, \Omega), \tag{3.1}$$

and to an expert policy $\boldsymbol{\pi}^E \in \Delta_A^S$. The aim of CIRL is to recover a reward $\boldsymbol{r}^*$ for which the expert policy $\boldsymbol{\pi}^E$ is optimal. A key observation is that such a reward may not be unique.

The *CIRL* problem with direct expert policy access (definition 3.1.1) makes assumptions that are not verified for most practical applications of *IRL*. The main motivation behind *IRL* is behavioural cloning, for example, from human demonstrations, a setting in which the expert policy is not directly accessible. This motivates a more practical definition of the *CIRL* in which the expert policy is accessed indirectly through a dataset of demonstrations sampled from the expert.

Problem Definition 3.1.2 (CIRL with expert dataset)**.** *In the CIRL problem we have access to: a regularized Constrained Markov Decision Processes (abbreviated CMDP) to which we have access through sampling only and for which we do not know the reward*

$$\textit{CMDP} \setminus \boldsymbol{r} = (S, A, P, \boldsymbol{\nu}, \Psi, \boldsymbol{b}, \gamma, \Omega), \tag{3.2}$$

a dataset $\mathcal{D}$ of trajectories produced by an expert policy $\boldsymbol{\pi}^E$, which we aim to clone:

$$\mathcal{D} = \{\tau_1, \tau_2, ..., \tau_N\}. \tag{3.3}$$

Each of these trajectories is a list of state-action pairs of length H:

$$\mathcal{D} = \{\tau_0, \tau_1, ..., \tau_N\} = \left\{\{(s_i^{(k)}, a_i^{(k)})\}_{k=0}^{H-1}\right\}_{i=0}^{N}, \tag{3.4}$$

where the pair $(s_i^{(k)}, a_i^{(k)}) \in S \times A$ denotes the state and actions at the k-th step of the i-th trajectory. The aim of CIRL is to recover a reward $\boldsymbol{r}^*$ for which the expert policy $\boldsymbol{\pi}^E$ is optimal. Such a reward may not be unique.

Note that the sampled expert policy setting of definition 3.1.2 indirectly provides access to the expert policy. To see this consider this simple estimator for the policy:

$$\hat{\boldsymbol{\pi}}^E(a|s) = \frac{\sum_{i=0}^{N}\sum_{k=0}^{H-1}\mathbb{1}(s_i^{(k)} = s, a_i^{(k)} = a)}{\sum_{i=0}^{N}\sum_{k=0}^{H-1}\mathbb{1}(s_i^{(k)} = s)}. \tag{3.5}$$

Where $\mathbb{1}$ denotes the indicator function. This observation allows us to reduce *CIRL* from expert dataset to *CIRL* with direct policy access. We can formalize our problem definition as finding a right-inverse solution map to the *CRL* solution map (definition 2.3.2).

Definition 3.1.1 (CIRL solution map)**.** *The solution map of the CIRL problem is a mapping $\mathit{CIRL} : \Delta_A^S \to \mathcal{R}$ which satisfies:*

$$(\mathit{CRL} \circ \mathit{CIRL})(\boldsymbol{\pi}^E) = \boldsymbol{\pi}^E. \tag{3.6}$$

A policy that satisfies (3.6) may not always exist. To ensure that it does, we introduce a key assumption, realizability.

Assumption 3.1.1 (Realizability)**.** *We assume that the expert policy $\boldsymbol{\pi}^E$ is optimal with respect to some reward $\boldsymbol{r}^E \in \mathcal{R}$:*

$$\boldsymbol{\pi}^E = \mathit{CRL}(\boldsymbol{r}^E). \tag{3.7}$$

3.2. Reformulation as an optimization problem

Having formalized exactly what problem we aim to solve, we will now discuss reducing the problem described in Section 3.1 into an optimization problem. In Proposition 3.2.1, we write out a minimax program that we claim solves the CIRL problem (in the sense that it satisfies definition 3.1.1).

Proposition 3.2.1 (Minmax program to solve CIRL)**.** *The optimal solution $\pi^*, \boldsymbol{r}^*$ returned by the minimax program:*

$$\begin{aligned} \min_{\boldsymbol{r}\in\mathcal{R}} \max_{\boldsymbol{\pi}\in\Delta_A^S} \; & J(\boldsymbol{\pi}, \boldsymbol{r}) - J(\boldsymbol{\pi}^E, \boldsymbol{r}) \\ \textit{s.t. } & \Psi^\top \boldsymbol{\mu}^{\boldsymbol{\pi}} \leq \boldsymbol{b} \end{aligned}, \tag{P1}$$

satisfies $\boldsymbol{\pi}^{\boldsymbol{\theta}^*} = \boldsymbol{\pi}^E$. Hence, the solutions of the program (P1) provide a mapping that satisfies the definition 3.1.1.

Proof. In the context of this proof, we introduce the feasible set:

$$\mathcal{F} = \left\{ \boldsymbol{\mu} \in \mathcal{M} \middle| \Psi^\top \boldsymbol{\mu} \leq \boldsymbol{b} \right\}, \tag{3.8}$$

which is the set of all valid occupancy measures that satisfy the Bellman flow constraints (Definition 2.1.10) and the *CMDP* constraints.

The first step of our proof is to make use of the one-to-one mapping between the occupancy measure and the policy (proposition 2.1.1) and to write the return functions $J(\boldsymbol{\pi}, \boldsymbol{r})$ and $J(\boldsymbol{\pi}^E, \boldsymbol{r})$ in their scalar product form, as a function of the occupancy $\boldsymbol{\mu}$ rather than policy-parameters $\boldsymbol{\theta}$:

$$J(\boldsymbol{\pi}, \boldsymbol{r}) = \langle \boldsymbol{r}, \boldsymbol{\mu} \rangle, -\beta\tilde{\Omega}(\boldsymbol{\mu}) \tag{3.9}$$

$$J(\boldsymbol{\pi}^E, \boldsymbol{r}) = \langle \boldsymbol{r}, \boldsymbol{\mu}^E \rangle - \beta\tilde{\Omega}(\boldsymbol{\mu}^E). \tag{3.10}$$

Using the occupancy measure form of J enables us to rewrite the program (P1) as follows:

$$\min_{\boldsymbol{r}\in\mathbb{R}} \max_{\boldsymbol{\mu}^{\boldsymbol{\pi}}\in\mathcal{F}} \; \langle \boldsymbol{r}, \boldsymbol{\mu} \rangle - \beta\tilde{\Omega}(\boldsymbol{\mu}) - \langle \boldsymbol{r}, \boldsymbol{\mu}^E \rangle + \beta\tilde{\Omega}(\boldsymbol{\mu}^E) \quad , \tag{P1.1}$$

where the decision variable $\boldsymbol{\mu}^{\boldsymbol{\pi}}$ is constrained to the feasible set $\mathcal{F}$ of the *CMDP*. For convenience, we choose to write our objective function as

$$f(\boldsymbol{\mu}, \boldsymbol{r}) = \langle \boldsymbol{r}, \boldsymbol{\mu} \rangle - \beta\tilde{\Omega}(\boldsymbol{\mu}) - \langle \boldsymbol{r}, \boldsymbol{\mu}^E \rangle + \beta\tilde{\Omega}(\boldsymbol{\mu}^E). \tag{3.11}$$

Observe that for any fixed $\boldsymbol{r} \in \mathcal{R}$ we have that:

$$\max_{\boldsymbol{\mu} \in \mathcal{F}} f(\boldsymbol{\mu}, \boldsymbol{r}) \geq 0. \tag{3.12}$$

This holds because:

1. either $\boldsymbol{\mu}^E$ maximizes the return for that reward, in which case $\max_{\boldsymbol{\mu} \in \mathcal{F}} f(\boldsymbol{\mu}, \boldsymbol{r}) = 0$,
2. or $\boldsymbol{\mu}^E$ doesn't, in which case $\max_{\boldsymbol{\mu} \in \mathcal{F}} f(\boldsymbol{\mu}, \boldsymbol{r}) > 0$.

Now completing the proof is done by observing that the lower-bound (3.12) is only ever achieved when $\boldsymbol{\mu}^E$ maximizes the return with respect to the reward $\boldsymbol{r}$, hence that minima are only achieved when *CIRL* is solved in the sense of definition 3.1.1. □

Proposition 3.2.1 provides a way to solve CIRL but is not implementable in practice, precisely the feasibility constraint $\Psi^\top \boldsymbol{\mu}^{\boldsymbol{\pi_\theta}} \leq \boldsymbol{b}$ explicitly makes use of the occupancy measure (which we most often do not have easy access to). One way to get around this is to relax the feasibility constraints out of the program (P1). This motivates working with a Lagrangian form such as:

$$\min_{\substack{\boldsymbol{r} \in \mathbb{R} \\ \boldsymbol{\lambda} \in \Lambda}} \max_{\boldsymbol{\theta} \in \mathbb{R}^p} J(\boldsymbol{\theta}, \boldsymbol{r}) - J(\boldsymbol{\theta}^E, \boldsymbol{r}) + \langle \boldsymbol{\lambda}, \boldsymbol{b} - \Psi^\top \boldsymbol{\mu}^{\boldsymbol{\pi_\theta}} \rangle \quad , \tag{P2}$$

where we introduce a penalty term $\langle \boldsymbol{\lambda}, \boldsymbol{b} - \Psi^\top \boldsymbol{\mu} \rangle$ that penalizes the constraint violation $\boldsymbol{b} - \Psi^\top \boldsymbol{\mu}$ up to a coefficient set by a Lagrangian multiplier vector $\boldsymbol{\lambda} \in \Lambda$. The set of admissible Lagrangian multipliers is given by:

$$\Lambda := \big\{ \boldsymbol{\lambda} \in \mathbb{R}^d | \lambda \geq 0 \big\}. \tag{3.13}$$

The optimization problem (P2) will be the main problem we will practically consider to solve *CIRL*. We name its objective function L since it is a Lagrangian:

$$L(\boldsymbol{\theta}, \boldsymbol{r}, \boldsymbol{\lambda}) = J(\boldsymbol{\theta}, \boldsymbol{r}) - J(\boldsymbol{\theta}^E, \boldsymbol{r}) + \langle \boldsymbol{\lambda}, \boldsymbol{b} - \Psi^\top \boldsymbol{\mu}^{\boldsymbol{\pi_\theta}} \rangle. \tag{3.14}$$

We abuse notation and equivalently write the objective in terms of policy parameters $\boldsymbol{\theta}$, of policy $\boldsymbol{\pi}$, or of occupancy measure $\boldsymbol{\mu}$. The occupancy measure form is especially useful as it is a concave-convex program with properties that we will exploit in our analysis:

$$L(\boldsymbol{\mu}, \boldsymbol{r}, \boldsymbol{\lambda}) = \langle \boldsymbol{r}, \boldsymbol{\mu} \rangle - \beta\tilde{\Omega}(\boldsymbol{\mu}) - \langle \boldsymbol{r}, \boldsymbol{\mu}^E \rangle + \beta\tilde{\Omega}(\boldsymbol{\mu}^E) + \langle \boldsymbol{\lambda}, \boldsymbol{b} - \Psi^\top \boldsymbol{\mu} \rangle. \tag{3.15}$$

We now define a useful function, the diminished reward $\tilde{\boldsymbol{r}}$, which in the light of strong duality, will be a key component for our analysis.

Definition 3.2.1 (Diminished reward)**.** *The function $\tilde{r} : S \times A \to \mathbb{R}$ defined as:*

$$\tilde{\boldsymbol{r}}_{\boldsymbol{\lambda}} = \boldsymbol{r} - \Psi\boldsymbol{\lambda}, \tag{3.16}$$

$$\tilde{r}(s, a) = r(s, a) - \langle \boldsymbol{\Psi}(s, a), \boldsymbol{\lambda} \rangle, \tag{3.17}$$

is called diminished reward.

Proposition 3.2.2 (Strong duality)**.** *Assuming that $\exists \boldsymbol{\mu} \in \mathcal{M}$ s.t. $\Psi^\top \boldsymbol{\mu}^E \leq \boldsymbol{b}$ and that $\boldsymbol{\mu} > 0$ (Slater's condition), and that the regularizer Ω associated with our CMDP is strictly convex, then the optimum of the Lagrangian dual (P2) is attained for some vector $\boldsymbol{\lambda}^* > 0$ and the optimal values found for $\boldsymbol{\theta}$ and $\boldsymbol{r}$ match the ones found by the primal. Furthermore, the solution of the program (P1) can be attained by solving an unconstrained program with diminished reward $\tilde{\boldsymbol{r}} = \boldsymbol{r} - \Psi\boldsymbol{\lambda}^*$:*

$$\mathit{CRL}(\boldsymbol{r}) = \mathit{RL}(\boldsymbol{r} - \Psi\boldsymbol{\lambda}^*) = \mathit{RL}(\tilde{\boldsymbol{r}}_{\boldsymbol{\lambda}^*}). \tag{3.18}$$

This is a classical convex optimization result. A proof can be found in Boyd and Vandenberghe 2004.

3.3. Performance metrics

To complete our setting Section, we discuss a few ways of relating the convergence of the optimization to the approximation error in *CIRL*. In an effort to provide a quantifiable measurement for safety, we introduce a metric that quantifies the extent to which the constraints are violated and thus provides a way to measure the safety of any policy *CIRL* solution.

Definition 3.3.1 (Constraint-violation)**.** *We measure constraint violation in the infinity norm, recall from Section 2.3 that the constraints are satisfied when $\boldsymbol{b} \geq \Psi^\top \boldsymbol{\mu_\theta}$, constraint-violation is thus naturally given as*

$$\max_{i \in d} \left[b_i - [\Psi^\top]_i \boldsymbol{\mu_\theta} \right]_+ = \| [\boldsymbol{b} - \Psi^\top \boldsymbol{\mu_\theta}]_+ \|_\infty, \tag{3.19}$$

Where $[x]_+ := \max\{x, 0\}$ is applied element wise and $[\Psi^\top]_i$ denotes the i-th collumn of the matrix Ψ. In other words this means that a bounded constraint violation $\| [\boldsymbol{b} - \Psi^\top(\boldsymbol{\theta})]_+ \|_\infty \leq \epsilon$ implies that no constraint is violated by more than ϵ.

Assuming we have a policy with bounded constraint violation, it is, easy to ensure that an approximate algorithm with bounded constraint violation always exactly satisfies the constraints by adding a safety margin on the constraint vector:

$$\boldsymbol{b}_{\text{margin}} = \boldsymbol{b} + \mathbf{1}\epsilon, \tag{3.20}$$

where $\mathbf{1}$ denotes the all-ones vector.

The algorithm that we will consider requires learning a policy; it therefore makes sense to provide a way of quantifying the quality of a policy with respect to the optimal solution for a given reward. We now study the suboptimality of policies for fixed reward function.

Definition 3.3.2 (Maximum Q-value suboptimality)**.** *Consider a reinforcement-learning problem with fixed reward $\boldsymbol{r}$ and $\boldsymbol{Q}^*$ the Q-function associated with any optimal solution to the problem, for any policy $\boldsymbol{\pi}$ we call the quantity*

$$\| \boldsymbol{Q}^{\boldsymbol{\pi}} - \boldsymbol{Q}^* \|_\infty = \max_{(s,a) \in S \times A} |Q^{\boldsymbol{\pi}}(s,a) - Q^*(s,a)|, \tag{3.21}$$

the maximum Q-value suboptimality, and we use it as a measure of the suboptimality of the policy. Any optimal policy $\boldsymbol{\pi}^*$ satisfies $\| \boldsymbol{Q}^{\boldsymbol{\pi}^*} - \boldsymbol{Q}^* \|_\infty = 0$.

Finally, we highlight that the way we will measure the quality of the discovered reward is directly through the suboptimality of the Lagrangian associated with a set of learned parameters $(\boldsymbol{\pi}, \boldsymbol{r}, \boldsymbol{\lambda})$:

$$\left| L(\boldsymbol{\pi}, \boldsymbol{r}, \boldsymbol{\lambda}) - L^* \right|, \tag{3.22}$$

where $L^* = L(\boldsymbol{\pi}, \boldsymbol{r}, \boldsymbol{\lambda})$ is the value taken by the Lagrangian upon convergence. We argue that this measurement is sensible because, by Proposition 3.2.1, we know that the Lagrangian suboptimality goes 0 when *CIRL* is solved. To see why this measurement makes sense, we first observe that assuming that we have exact access to the expert policy parameters $L^* = 0$ we thus have:

$$\begin{aligned}
\left| L(\boldsymbol{\pi}, \boldsymbol{r}, \boldsymbol{\lambda}) - L^* \right| &= \left| L(\boldsymbol{\pi}, \boldsymbol{r}, \boldsymbol{\lambda}) \right| && (3.23)\\
&= J(\boldsymbol{\theta}, \boldsymbol{r}) - J(\boldsymbol{\theta}^E, \boldsymbol{r}) + \langle \boldsymbol{\lambda}, \boldsymbol{b} - \Psi^\top \boldsymbol{\mu}^{\boldsymbol{\pi_\theta}} \rangle && (3.24)\\
&= J(\boldsymbol{\theta}, \boldsymbol{r}) - J(\boldsymbol{\theta}^*_r, \boldsymbol{r}) + J(\boldsymbol{\theta}^*_r, \boldsymbol{r}) - J(\boldsymbol{\theta}^E, \boldsymbol{r}) + \langle \boldsymbol{\lambda}, \boldsymbol{b} - \Psi^\top \boldsymbol{\mu}^{\boldsymbol{\pi_\theta}} \rangle && (3.25)\\
&\geq J(\boldsymbol{\theta}, \boldsymbol{r}) - J(\boldsymbol{\theta}^*_r, \boldsymbol{r}) + J(\boldsymbol{\theta}^*_r, \boldsymbol{r}) - J(\boldsymbol{\theta}^E, \boldsymbol{r}), && (3.26)
\end{aligned}$$

Where the last line uses that constraint violation is positive. Rearranging we get:

$$\begin{aligned}
\left| L(\boldsymbol{\pi}, \boldsymbol{r}, \boldsymbol{\lambda}) \right| + J(\boldsymbol{\theta}^*_r, \boldsymbol{r}) - J(\boldsymbol{\theta}, \boldsymbol{r}) &\geq J(\boldsymbol{\theta}^*_r, \boldsymbol{r}) - J(\boldsymbol{\theta}^E, \boldsymbol{r}) && (3.27)\\
&= \beta \mathbb{E}_{s \sim \boldsymbol{\mu}^{\boldsymbol{\pi}^E}_S} \left[D_{\mathsf{KL}}\big(\boldsymbol{\pi}^E(\cdot|s) || \boldsymbol{\pi}^*_{\boldsymbol{r}}(\cdot|s) \big) \right], && (3.28)
\end{aligned}$$

where the last equality holds by Lemma 2.4.3 (soft-suboptimality). So in plain English, if the Lagrangian is bounded and the learned policy's suboptimality with respect to the reward $\boldsymbol{r}$ is bounded, then optimal policy associated with the learned $\boldsymbol{r}$ is close to the expert policy.

4

An algorithm to solve CIRL in the exact gradient setting

In the following Chapter, we introduce the main algorithm studied in the thesis and prove that it converges globally in a finite iteration count (in $O(1/\epsilon^2)$ time) under the assumption that we have access to an exact-gradient oracle. Except when specified otherwise, results in this section are new contributions.

4.1. Designing an algorithm to solve CIRL

Recall that one way of solving *CIRL* in the sense of Definition 3.1.1 is by solving the minimax problem:

$$\min_{\substack{\boldsymbol{r}\in\mathbb{R}\\ \boldsymbol{\lambda}\in\Lambda}} \max_{\boldsymbol{\theta}\in\mathbb{R}^{nm}} L(\boldsymbol{\theta}, \boldsymbol{r}, \boldsymbol{\lambda}), \tag{P2}$$

this can equivalently be solved by running gradient ascent on the occupancy measure $\boldsymbol{\mu}$ or on the policy $\boldsymbol{\pi}$. The occupancy-measure Lagrangian $L(\boldsymbol{\mu}, \boldsymbol{r}, \boldsymbol{\lambda})$ has arguably more appealing properties than its policy/parameters counterpart $L(\boldsymbol{\theta}, \boldsymbol{r}, \boldsymbol{\lambda})$ as L is concave with respect to $\boldsymbol{\mu}$ but not with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$). Regardless, we will not devise an algorithm that runs gradient ascent on the occupancy measure vector $\boldsymbol{\mu}$, but rather on the policy parameters θ. This is for several reasons:

1. algorithms directly optimizing with the occupancy measure as a decision variable are quite impractical. The main reason for this is that the set of valid occupancy measures $\mathcal{M}$ can only be computed when the MDP dynamics are known.
2. If we explicitly run the algorithm on the occupancy measure itself, then the algorithm needs to work with vectors $\boldsymbol{\mu} \in \mathbb{R}^{nm}$ as big as the state-action space. This defeats one of the main purposes of reinforcement learning, which is learning policies in very large state-action spaces.

Furthermore, recent successes in proving global convergence of policy gradient algorithms [Cen et al. 2021; Mei et al. 2020; Agarwal et al. 2020] motivate the choice of devising such an algorithm. Specifically, [Cen et al. 2021] has shown that Natural Policy Gradients display a linear convergence rate when regularized with Shannon's entropy.

Therefore we propose solving the *CIRL* problem with gradient descent on the variables $\boldsymbol{\lambda}$ and $\boldsymbol{r}$ and with *natural policy gradient* ascent (*NPG*) on the parameters $\boldsymbol{\theta}$ of the policy $\boldsymbol{\pi}$. This formulation closely matches primal-dual methods proposed for *CRL* by [D. Ding et al. 2020] (but differs in the sense that it makes use of entropy regularization, which modifies the convergence dynamics in the primal) and for [Zeng et al. 2022] but differs in the sense that we propose using projected gradient descent on the reward class.

For any practical implementation, we will have to devise an algorithm that works by sampling the MDP under consideration. But we will first restrict our analysis to the "*exact-gradients*" setting, i.e.

we will assume access to exact gradient oracles for our algorithm. We will also assume direct expert policy access, as specified in Problem Definition 3.1.1.

In the following section, we introduce the algorithm more formally.

4.2. The algorithm, NPG-CIRL

We describe our algorithm in a setting where both the reward and the policy are set by parameter vectors. We let $\boldsymbol{w} \in \mathbb{R}^f$ be the parameter vector of the reward $\boldsymbol{r} \in \mathcal{R} \subset \mathbb{R}^{nm}$ and $\boldsymbol{\theta} \in \mathbb{R}^{nm}$ be the parameter vector of the policy $\boldsymbol{\pi} \in \Delta_A^S$. From now on, we will use notations $\boldsymbol{r_w}$ and $\boldsymbol{\pi_\theta}$ to denote the reward and policies parameterized by their respective parameter vectors. This allows the specification of our algorithm to be more general in the sense that our algorithm description encompasses everything from direct parameterization to neural network parameterization. Later, when analyzing our algorithm, we will explicitly specify which parameterization we consider.

Algorithm 2: NPG-CIRL

Set the learning rates $\eta_\theta = \frac{1}{\beta}, \eta_z = \frac{1}{\sqrt{T}}$
Initialize the algorithm at some point $(\boldsymbol{w}^{(0)}, \boldsymbol{\lambda}^{(0)}, \boldsymbol{\theta}^{(0)})$
foreach *iteration* $t = 0, 1, ..., T-1$ **do**
  $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} + \eta_\theta (\mathfrak{F}^{\boldsymbol{\theta}})^\dagger \nabla_{\boldsymbol{\theta}} L(\boldsymbol{w}^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{\theta}^{(t)})$
  $\boldsymbol{w}^{(t+1)} \leftarrow \mathcal{P}_{\mathsf{dom}(\boldsymbol{w})}\big(\boldsymbol{r}_{\boldsymbol{w}^{(t}} - \eta_z \nabla_{\boldsymbol{w}} L(\boldsymbol{w}^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{\theta}^{(t)})\big)$
  $\boldsymbol{\lambda}^{(t+1)} \leftarrow \mathcal{P}_{\boldsymbol{\lambda} \in \Lambda}\big(\boldsymbol{\lambda}^{(t)} - \eta_z \nabla_{\boldsymbol{\lambda}} L(\boldsymbol{w}^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{\theta}^{(t)})\big)$
end
return $(\boldsymbol{w}^{(T)}, \boldsymbol{\lambda}^{(T)}, \boldsymbol{\theta}^{(T)})$.

The algorithm, which we call *NPG-CIRL* works by running simultaneous gradient descent and ascent steps on all three variables it optimizes. Note that the learning rate choice is actually quite practical to pick as β is a regularization factor which is set arbitrarily, and the learning rate in η_z is a function of the total number of algorithm steps T. It also requires the use of a projection operator to ensure the Lagrangian multiplier stays in its allowed domain Λ, but this projection is trivial as it is given in closed form by:

$$[\mathcal{P}_\Lambda(\boldsymbol{\lambda})]_i = \mathsf{max}\{[\boldsymbol{\lambda}]_i, 0\}, \ \forall i \in [d]. \tag{4.1}$$

Finally, we must underline that the computation of the policy step makes use of the Moore-Penrose Inverse of the *Fisher information matrix* (*FIM*) $(\mathfrak{F}^\theta)^\dagger$ which is a computationally expansive quantity to compute. This drawback can be mitigated in practice by only computing matrix-vector products and by using the conjugate gradient method (as is most notably done in TRPO [Schulman, Levine, et al. 2015]). In this work, we will always assume that we have direct access to the *FIM*.

We will specifically analyze Algorithm 2 in a setting where the policy is tabular and softmax parameterized, where the reward is restricted to linear reward class $\mathcal{R}_{L_1}^\Phi$ and we use negative Shannon entropy (Definition 2.2.7) as our regularizer. We formalize these three facts with three assumptions.

Assumption 4.2.1 (Tabular softmax policy parameterization)**.** *Our policy $\boldsymbol{\pi}^{\boldsymbol{\theta}}$ is parameterized by the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{nm}$. The parameterization is softmax, i.e. for any state action pair $(s, a) \in S \times A$, we have:*

$$\pi^{\boldsymbol{\theta}}(a|s) = \frac{\mathsf{exp}(\theta(s,a))}{\sum_{a' \in A} \mathsf{exp}\, \theta(s,a')}, \ \forall (s,a) \in S \times A. \tag{4.2}$$

Assumption 4.2.2 (Linear reward class)**.** *Our reward class $\mathcal{R}_{L_1}^\Phi$ is given by:*

$$\mathcal{R}_{L_1}^\Phi := \big\{ \boldsymbol{r^w} = \Phi \boldsymbol{w} \big| \boldsymbol{w} \in \mathbb{R}^k, \ \Phi \in \mathbb{R}^{nm \times k}, \ \|\boldsymbol{w}\|_1 \leq 1 \big\}. \tag{4.3}$$

Assumption 4.2.3 (Negative shannon regularizer)**.** *We let our regularizer $\Omega : \Delta_A \to \mathbb{R}$ be the negative Shannon entropy (Definition 2.2.7):*

$$\Omega(\boldsymbol{\pi}(\cdot|s)) = -H(\boldsymbol{\pi}(\cdot|s)). \tag{4.4}$$

To ensure that strong duality holds (Proposition 3.2.2), we will also require that the occupancy measure is always bounded away from 0. We also formalize this as an assumption.

Assumption 4.2.4 (Non-vanishing occupancy measure)**.** *For any $(s, a) \in S \times A$, it holds that:*

$$\mu(s, a) > 0. \tag{4.5}$$

4.3. Analysis

In the following Section, we analyze Algorithm 2 in the tabular setting with softmax policy parameterization and a linear reward class on a discrete, infinite horizon *CMDP*. We show that with access to exact gradient oracles, Algorithm 2 solves Problem 3.1.1 and converges globally at a rate of $O(1/\sqrt{T})$. This is equivalent to showing that we can get to an arbitrarily small ϵ error in $O(1/\epsilon^2)$ iterations. This result is formalized in Theorem 4.3.1, in which we express convergence in terms of suboptimality of the Lagrangian.

4.3.1. Setting for the analysis, and formal statement of main results

Before introducing any new assumption to our setting we re-state the Lagrangian that we will optimize on:

$$L(\boldsymbol{\theta}, \boldsymbol{r}, \boldsymbol{\lambda}) = J(\boldsymbol{\theta}, \boldsymbol{r}) - J(\boldsymbol{\theta}^E, \boldsymbol{r}) + \langle \boldsymbol{\lambda}, \boldsymbol{b} - \Psi^\top \boldsymbol{\mu}^{\boldsymbol{\pi_\theta}} \rangle. \tag{4.6}$$

$$= \langle \boldsymbol{r}, \boldsymbol{\mu}^{\boldsymbol{\pi_\theta}} \rangle - \beta\tilde{\Omega}(\boldsymbol{\mu}^{\boldsymbol{\pi_\theta}}) - \langle \boldsymbol{r}, \boldsymbol{\mu}^E \rangle + \beta\tilde{\Omega}(\boldsymbol{\mu}^E) + \langle \boldsymbol{\lambda}, \boldsymbol{b} - \Psi^\top \boldsymbol{\mu}^{\boldsymbol{\pi_\theta}} \rangle. \tag{4.7}$$

Because of the one-to-one mapping between policy and occupancy measures (Proposition 2.1.1), the two forms (4.6) and (4.7) are completely equivalent. Writing the Lagrangian in form (4.7) allows for an easier derivation of the gradient of the dual variables $\boldsymbol{w}$ and $\boldsymbol{\lambda}$.

We use the notation $\boldsymbol{\pi_\theta} \in \Delta_A^S$ to denote the policy parameterized by the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{nm}$. The notation $\theta(s, a)$ refers to the element of the vector $\boldsymbol{\theta}$ associated with the state action pair (s, a) in the same way we would write $\pi(a|s)$ for the element of the policy-vector associated with said state-action pair. Similarly, we denote the reward parameterized by reward-parameterization vector $\boldsymbol{w} \in \mathbb{R}^k$ as $\boldsymbol{r_w} \in \mathcal{R}_\Phi^{L_1}$. We write $\boldsymbol{\theta}^{(t)}$, $\boldsymbol{w}^{(t)}$ and $\boldsymbol{\lambda}^{(t)}$ to denote the decision variables of Algorithm 2 after t iterations, and $\boldsymbol{\theta}^{(T)}$, $\boldsymbol{w}^{(T)}$ and $\boldsymbol{\lambda}^{(T)}$ to denote their value at the last iteration. For convenience, we use $\boldsymbol{\pi}^{(t)} = \boldsymbol{\pi}_{\boldsymbol{\theta}^{(t)}}$ to denote the policy at the t-th iteration and $\boldsymbol{\mu}^{(t)} = \boldsymbol{\mu}^{\boldsymbol{\pi}_{\boldsymbol{\theta}^{(t)}}}$ to denote the occupancy measure at that iteration. We use $\boldsymbol{\pi}^E$ to denote the expert policy, $\boldsymbol{\mu}^E$ the occupancy measure that it induces, $\boldsymbol{\theta}^E$ is a parameterization that induces it. We write $\boldsymbol{\pi}_{\boldsymbol{r}}^* = \mathsf{CRL}(\boldsymbol{r})$ to denote the optimal policy for some reward $\boldsymbol{r} \in \mathbb{R}$. The scalar L^* is the value that the Lagrangian takes when it has reached a saddle point.

Under Assumption 4.2.2, the gradients used for the descent steps of Algorithm 2[1] are given by:

$$\nabla_{\boldsymbol{w}} L(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\lambda}) = \Phi^\top(\boldsymbol{\mu}^{\boldsymbol{\pi_\theta}} - \boldsymbol{\mu}^E), \tag{4.8}$$

$$\nabla_{\boldsymbol{\lambda}} L(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\lambda}) = \boldsymbol{b} - \Psi^\top \boldsymbol{\mu}^{\boldsymbol{\pi_\theta}}. \tag{4.9}$$

The projected gradient descent steps on variables $\boldsymbol{w}$ and $\boldsymbol{\lambda}$ are essentially identical. We thus concatenate $\boldsymbol{w}$ and $\boldsymbol{\lambda}$ together in a variable $\boldsymbol{z} = [\boldsymbol{w}^\top, \boldsymbol{\lambda}^\top]^\top$ for the purpose of the analysis. This allows only to consider two simultaneous steps:

$$\boldsymbol{z}^{(t+1)} \leftarrow \mathcal{P}_{\boldsymbol{z} \in \mathsf{dom}(\boldsymbol{z})}\left(\boldsymbol{z}^{(t)} - \eta_z \nabla_{\boldsymbol{z}} L(\boldsymbol{z}^{(t)}, \boldsymbol{\theta}^{(t)})\right) \tag{4.10}$$

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} + \eta_\theta (\mathfrak{F}^{\boldsymbol{\theta}})^\dagger \nabla_{\boldsymbol{\theta}} L(\boldsymbol{z}^{(t)}, \boldsymbol{\theta}^{(t)}), \tag{4.11}$$

and when considering projected steps, we use the following notation convention for the analysis:

$$\boldsymbol{z}^{(t+1)} \leftarrow \mathcal{P}_{\boldsymbol{z} \in \mathsf{dom}(\boldsymbol{z})}\left(\boldsymbol{z}^{(t+1/2)}\right) \tag{4.12}$$

$$\boldsymbol{z}^{(t+1/2)} = \boldsymbol{z}^{(t)} - \eta_z \nabla_{\boldsymbol{z}} L(\boldsymbol{z}^{(t)}, \boldsymbol{\theta}^{(t)}). \tag{4.13}$$

Where $\mathsf{dom}(\boldsymbol{z}) = \mathsf{dom}(\boldsymbol{w}) \times \Lambda$. We now formally state our global convergence result.

[1] These gradients are derived from equation (4.7).

Theorem 4.3.1 (Global convergence of NPG-CIRL)**.** *Consider the sequence of policy parameters $\{\boldsymbol{\theta}^{(t)}\}_{t=1}^{T}$, of reward parameters $\{\boldsymbol{w}^{(t)}\}_{t=1}^{T}$, and of Lagrangian multipliers $\{\boldsymbol{\lambda}^{(t)}\}_{t=1}^{T}$ generated by running T steps of Algorithm 2 with policy-learning rate $\eta_\theta = 1/\beta$, learning rate $\eta_z = \frac{1}{\sqrt{T}}$ and when assumptions 4.2.1 (tabular softmax policy), 4.2.2 (linear reward class), 4.2.3 (negative shanon regularized) and 4.2.4 (non vanishing occupancy measure) are satisfied. The minimum Lagrangian suboptimality attained in the sequence satisfies:*

$$\min_{1\leq i\leq T} \left|L(\boldsymbol{\theta}^{(i)}, \boldsymbol{w}^{(i)}, \boldsymbol{\lambda}^{(i)}) - L^*\right| = O\Big(\frac{1}{\sqrt{T}}\Big). \tag{4.14}$$

Another key result we must state is that the reward parameterization $\boldsymbol{w}^{(T)}$ returned after T iterations is associated with a policy that satisfies bounded constraint violation, specifically we have that constraint violation goes down at a rate $O(1/\sqrt{T})$. We formalize that result through Lemma 4.3.2.

Lemma 4.3.2 (Constraint violation upper bound)**.** *Consider the iteration $(\boldsymbol{\theta}^{(T)}, \boldsymbol{w}^{(T)}, \boldsymbol{\lambda}^{(T)})$, returned by Algorithm 2, and when assumptions 4.2.1 (tabular softmax policy), 4.2.2 (linear reward class), 4.2.3 (negative shanon regularized) and 4.2.4 (non vanishing occupancy measure) are satisfied. Then we have that constraint violation satisfies:*

$$\|[\boldsymbol{b} - \Psi^\top \boldsymbol{\mu}^{(T)}]_+\|_\infty = O\Big(\frac{1}{\sqrt{T}}\Big). \tag{4.15}$$

The proof requires results that we will show during the analysis of the algorithm and is thus deferred to the appendix, section B.4.

4.3.2. The big picture, intuition for the analysis of the algorithm

The main idea behind the analysis of the algorithm is to look at it through the lens of *dual-descent*. We will study the dual of the minimax optimization problem (P2),

$$\min_{\substack{\boldsymbol{r}\in\mathcal{R}\\ \boldsymbol{\lambda}\in\Lambda}} \max_{\boldsymbol{\theta}\in\mathbb{R}^{nm}} L(\boldsymbol{\theta}, \boldsymbol{r}, \boldsymbol{\lambda}), \tag{P2}$$

which we will denote (D) and which we define as:

$$D(\boldsymbol{w}, \boldsymbol{\lambda}) = \sup_{\boldsymbol{\theta}\in\mathbb{R}^{nm}} L(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\lambda}), \tag{4.16}$$

Observe that solving the minimization problem

$$\min_{\substack{\boldsymbol{r}_{\boldsymbol{w}}\in\mathcal{R}\\ \boldsymbol{\lambda}\in\Lambda}} D(\boldsymbol{w}, \boldsymbol{\lambda}), \tag{D1}$$

is completely equivalent to solving the minimax problem (P2). The main idea of our analysis (Figure 4.1) is to use the fast convergence properties of NPG to ensure that Algorithm 2 converges quickly to (and then stays contained within) a neighbourhood around a *locally optimal* $\boldsymbol{\theta}^{(t)}$. We formally show that this is true in Lemma 4.3.3. That notion of local optimality means that the iterations $(\boldsymbol{\theta}^{(t)}, \boldsymbol{w}^{(t)}, \boldsymbol{\lambda}^{(t)})$ generated by Algorithm 2 provide an approximation of the dual function $D(\boldsymbol{r}^{(t)}, \boldsymbol{\lambda}^{(t)})$. This, in turn, facilitates our analysis as it enables us to analyze the convergence of gradient descent on the simpler problem (D1). We then show the convergence of the Algorithm by checking that its iterations provide an approximation of gradient descent on the dual function D. This is the main idea behind the proof of Theorem 4.3.1. Analyzing the convergence of gradient descent on problem D1 is made easy because convexity is ensured in the dual, which we formalize in the following proposition.

Proposition 4.3.1 (Dual convexity)**.** *[Boyd and Vandenberghe 2004] Consider the function $f : X \times Y \to \mathbb{R}$ which is convex in X and concave in Y, its dual $d : X \to \mathbb{R}$, which is defined for any $x \in X$ as:*

$$d(\boldsymbol{x}) = \sup_{\boldsymbol{y}\in Y} f(\boldsymbol{x}, \boldsymbol{y}). \tag{4.17}$$

is convex.

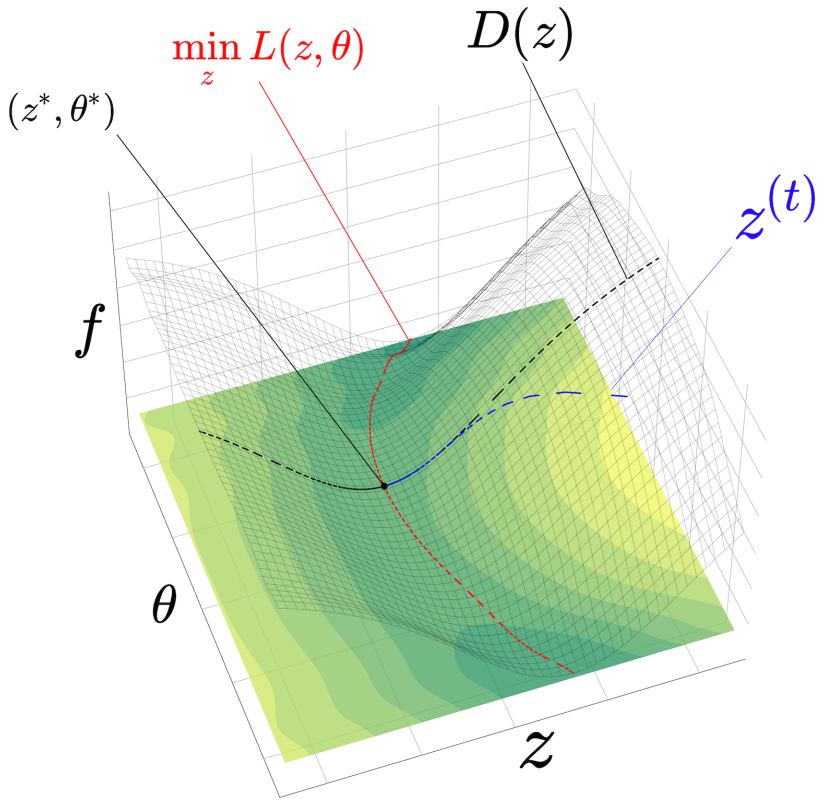


Figure 4.1: A representation of the process behind the convergence of Algorithm 2. The algorithm maximizes the Lagrangian L with respect to $\boldsymbol{\theta}$ while minimizing it with respect to $\boldsymbol{z}$. Our analysis relies on the observation that the algorithm converges fast (at a linear rate) to a neighborhood of the dual function D which is convex in $\boldsymbol{z}$.

We now formally state that the dual D is convex.

Proposition 4.3.2 (The dual D is convex.)**.** *Assuming that the reward parameterization is linear, the dual function*

$$D(\boldsymbol{w}, \boldsymbol{\lambda}) = \sup_{\boldsymbol{\theta} \in \mathbb{R}^{nm}} L(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\lambda}), \tag{4.18}$$

is convex with respect to the variables $\boldsymbol{w}$ and $\boldsymbol{\lambda}$.

Proof. Dual convexity holds because the Lagrangian is convex in the variables $\boldsymbol{w}$ and $\boldsymbol{\lambda}$. This can be seen by explicitly writing it out,

$$\begin{aligned} L(\boldsymbol{\theta}, \boldsymbol{r}, \boldsymbol{\lambda}) &= \langle \boldsymbol{r}, \boldsymbol{\mu}^{\boldsymbol{\pi_\theta}} \rangle - \beta \tilde{\Omega}(\boldsymbol{\mu}^{\boldsymbol{\pi_\theta}}) - \langle \boldsymbol{r}, \boldsymbol{\mu}^E \rangle + \beta \tilde{\Omega}(\boldsymbol{\mu}^E) + \langle \boldsymbol{\lambda}, \boldsymbol{b} - \Psi^\top \boldsymbol{\mu} \rangle & (4.19) \\ &= \langle \Phi \boldsymbol{w}, \boldsymbol{\mu}^{\boldsymbol{\pi_\theta}} - \boldsymbol{\mu}^E \rangle - \beta \tilde{\Omega}(\boldsymbol{\mu}^{\boldsymbol{\pi_\theta}}) + \beta \tilde{\Omega}(\boldsymbol{\mu}^E) + \langle \boldsymbol{\lambda}, \boldsymbol{b} - \Psi^\top \boldsymbol{\mu} \rangle, & (4.20) \end{aligned}$$

and observing that it is a sum of convex functions. The Lagrangian $L(\boldsymbol{\mu}, \boldsymbol{r}, \boldsymbol{\lambda})$, being concave-convex, we can apply Proposition 4.3.1 and see that:

$$D_{\boldsymbol{\mu}}(\boldsymbol{w}, \boldsymbol{\lambda}) = \max_{\boldsymbol{\mu} \in \mathcal{M}}. \tag{4.21}$$

Observing that:

1. there exists a one-to-one mapping between policy and occupancy measure (Proposition 2.1.1),

2. there exists a one-to-one mapping between the softmax policy and the policy parameter vector $\boldsymbol{\theta}$,

we see that there exists a *one-to-one mapping between policy parameters and occupancy*. And thus we have

$$\sup_{\boldsymbol{\mu}\in\mathcal{M}} L(\boldsymbol{\mu}, \boldsymbol{w}, \boldsymbol{\lambda}) = \sup_{\boldsymbol{\theta}\in\mathbb{R}^{nm}} L(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\lambda}), \tag{4.22}$$

and the proposition is verified. □

4.3.3. Convergence to a local optimum

We now get to the core of our analysis. We prove that our algorithm converges to a locally optimal policy $\boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}^{(t)}}$. By locally optimal, we mean optimal with respect to the reward parameterized by $\boldsymbol{w}^{(t)}$ and the cost induced by the Lagrange multiplier $\boldsymbol{\lambda}^{(t)}$, set at time t by the iterations produced by Algorithm 2. In order to spare ourselves the complexity of dealing with separate terms for reward and cost, we will study the convergence of our algorithm using diminished rewards (see Definition 3.2.1):

$$\tilde{\boldsymbol{r}}_{\boldsymbol{\lambda}} = \boldsymbol{r} - \Psi\boldsymbol{\lambda}. \tag{4.23}$$

Essentially our approach will be to follow an analysis similar to the usual analysis of entropy-regularized *NPG* (see [Cen et al. 2021]). As in [Cen et al. 2021] we will use γ-contraction property of the operator $\mathcal{T}_\beta$. Where analysis differs from that of [Cen et al. 2021] is in showing that even if the diminished reward and Lagrange multiplier are regularly updated, the policy is guaranteed to converge to a neighbourhood of the locally optimal policy $\boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}^{(t)}}$. In other words, the policy converges "faster" than the reward and cost terms "move". Showing that small reward changes yield small perturbations in the policy updates be achieved by showing that two key propositions which establish that Q-values are Lipschitz with respect to the reward parameters $\boldsymbol{\lambda}$ and Lagrange multiplier $\boldsymbol{\lambda}$.

Proposition 4.3.3 (Q-values are Lipschitz w.r.t diminished reward for fixed policy)**.** *Consider two diminished rewards $\tilde{\boldsymbol{r}}^1$ and $\tilde{\boldsymbol{r}}^2$ induced by two reward parameters and Lagrange multipliers $\boldsymbol{w}_1$, $\boldsymbol{\lambda}_1$ and $\boldsymbol{w}_2$, $\boldsymbol{\lambda}_2$. Assume that the gradient of the reward with respect to the parameters is upper bounded by $\|\Phi\|$[2]. We have, in the setting of a regularized, constrained MDP, that the following bound on the difference between Q-values induced by the same policy $\boldsymbol{\pi} \in \Delta_A^S$ on the different diminished rewards $\tilde{\boldsymbol{r}}^1$ and $\tilde{\boldsymbol{r}}^2$ holds:*

$$\|\boldsymbol{Q}^{\boldsymbol{\pi}}_{\tilde{\boldsymbol{r}}^1} - \boldsymbol{Q}^{\boldsymbol{\pi}}_{\tilde{\boldsymbol{r}}^2}\|_\infty \leq \|\Psi\|\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2 + \|\Psi\|\|\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2\|_2 \tag{4.24}$$

$$\leq C_z\|\boldsymbol{z}_1 - \boldsymbol{z}_2\|, \tag{4.25}$$

where $C_z = 2(\|\Phi\| + \|\Psi\|)$ and $\boldsymbol{z} = [\boldsymbol{w}^\top, \boldsymbol{\lambda}^\top]^\top$. The proof of this proposition is deferred to Appendix B.2.

Proposition 4.3.4 (Optimal Q-values are Lipschitz w.r.t diminished reward)**.** *Consider two diminished rewards $\tilde{\boldsymbol{r}}^1$ and $\tilde{\boldsymbol{r}}^2$ induced by two reward parameters and Lagrange multipliers $\boldsymbol{w}_1$, $\boldsymbol{\lambda}_1$ and $\boldsymbol{w}_2$, $\boldsymbol{\lambda}_2$. Assume that the gradient of the reward with respect to the parameters is upper bounded by $\|\Phi\|$[3]. We have, in the setting of a regularized, constrained MDP, that the following bound on the difference between the optimal Q-values induced by both diminished rewards holds:*

$$\|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^2} - \boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^2}\|_\infty \leq \|\Psi\|\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2 + \|\Psi\|\|\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2\|_2 \tag{4.26}$$

$$\leq C_z\|\boldsymbol{z}_1 - \boldsymbol{z}_2\|, \tag{4.27}$$

where $C_z = 2(\|\Phi\|^\top + \|\Psi\|^\top)$ and $\boldsymbol{z} = [\boldsymbol{w}, \boldsymbol{\lambda}]^\top$, the second line provides a more convenient, less tight bound which doesn't distinguish between both parameter vectors $\boldsymbol{w}$ and $\boldsymbol{\lambda}$. This result is a corollary of Proposition 4.3.3, and its proof is deferred to Appendix B.2.

[2] Same remark as in Proposition 4.3.4.

[3] This is an abuse of notation, when considering a linear reward-parameterization, the gradients of the reward w.r.t the reward-parameters are upper bounded in $\|\cdot\|_2$ norm by the spectral norm of the reward feature matrix, which we indeed denote $\|\Phi\|$ but we might pick another parameterization for the rewards. In which case, the result we show still applies, but $\|\Phi\|$ just denotes the upper bound on the $\|\cdot\|_2$ norm of the gradient of the reward parameterization.

Next, let us state a useful proposition that relates convergence in the Q-values to convergence in the policy.

Proposition 4.3.5 (Log policy-error is upper bounded by maximum Q-value suboptimality)**.** *[Cen et al. 2021] Consider the sequence of policies $\{\boldsymbol{\pi}^{(t)}\}_{t=1}^{T}$ and of Q-values $\{\boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}\}$ generated by running T steps of NPG-CIRL (with policy-learning rate $\eta_\theta = 1/\beta$ and dual learning rate $\eta_z = \frac{1}{T^u}$). Then we have that:*

$$\|\log \boldsymbol{\pi}^{(t+1)} - \log \boldsymbol{\pi}^{*}_{\tilde{\boldsymbol{r}}^{(t)}}\|_\infty \leq 2\|\boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(t)}}\|_\infty, \tag{4.28}$$

where $\tilde{\boldsymbol{r}}^{(t)}$ denotes the diminished reward (Definition 3.2.1) associated with reward parameters $\boldsymbol{w}^{(t)}$ and Lagrange multipler $\boldsymbol{\lambda}^{(t)}$. On the other hand $\boldsymbol{\pi}^{*}_{\tilde{\boldsymbol{r}}^{(t)}}$ denotes the policy optimal with respect to the reward $\boldsymbol{r}^{(t)}$ that satisfies the constraints $\Psi^\top \boldsymbol{\mu}^{\boldsymbol{\pi_\theta}} \leq \boldsymbol{b}$. In other words, $\boldsymbol{\pi}^{*}_{\tilde{\boldsymbol{r}}^{(t)}} = \text{CRL}(\tilde{\boldsymbol{r}}^{(t)})$ is the policy optimal with respect to the diminished reward $\tilde{\boldsymbol{r}}^{(t)}$ at time t. The proof of this proposition is deferred to appendix B.1.

Equipped with these three propositions, we are ready to state and prove our main local-convergence result (Lemma 4.3.3).

Lemma 4.3.3 (Local convergence of NPG-CIRL with exact gradients)**.** *We consider the sequences $\{\tilde{\boldsymbol{r}}^{(t)}\}_{t=0}^{T}$ and $\{\pi^{(t)}\}_{t=0}^{T}$ of diminished rewards and policies generated by Algorithm 2, with tabular softmax policy parameterization (as in (4.2)) and a linear reward class (as in (4.3)) the policy converges to a local optimum (to a policy optimal w.r.t. $\tilde{\boldsymbol{r}}^{(t)}$) at rate:*

$$\|\boldsymbol{Q}^{(T)}_{\tilde{\boldsymbol{r}}^{(T)}} - \boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(T)}}\|_\infty \leq 2\|\Phi\|_1\gamma^T + 2C'_z\frac{1-\gamma^T}{1-\gamma}\frac{1}{T^u}, \tag{4.29}$$

$$\|\log \boldsymbol{\pi}^{(T+1)} - \log \boldsymbol{\pi}^{*}_{\tilde{\boldsymbol{r}}^{(T)}}\|_\infty \leq 4\|\Phi\|_1\gamma^T + 4C'_z\frac{1-\gamma^T}{1-\gamma}\frac{1}{T^u}, \tag{4.30}$$

where $C_z = 2(\|\Phi\| + \|\Psi\|)(\|\Phi\| + \|\Psi\| + \|\boldsymbol{b}\|_2)$, $\|\Phi\|_1$ is the maximum column sum matrix norm of the feature matrix Φ and $\boldsymbol{\pi}^{*}_{\tilde{\boldsymbol{r}}^{(T)}}$ is the policy optimal with respect to the diminished reward $\tilde{\boldsymbol{r}}^{(t)}$.

Proof. Leveraging Proposition 4.3.5 we restrict our study of policy convergence to that of $\boldsymbol{Q}$-value convergence. To do so we start by decomposing our Q-value error as follows:

$$\|\boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(t)}}\|_\infty = \|\boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(t)}} \overbrace{+\boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(t-1)}} - \boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(t-1)}}}^{=0} + \overbrace{\boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t-1)}} - \boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t-1)}}}^{=0}\|_\infty \tag{4.31}$$

$$\overset{(i)}{\leq} \overbrace{\|\boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t-1)}} - \boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(t-1)}}\|_\infty}^{(A)} + \overbrace{\|\boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(t-1)}}\|_\infty}^{(B)} + \overbrace{\|\boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t-1)}}\|_\infty}^{(C)}. \tag{4.32}$$

In (i) we just rearrange the terms and take a triangle inequality. This leaves us with three terms (A), (B) and (C) that we need to bound to show convergence. Observe that (A) matches the error term on the left hand side of the inequality but at iteration $(t-1)$, this suggests a recursion. Terms (B) and (C) are similar in that in both case they are a function of the diminished reward-step, but not the policy step. We start by bounding terms (B) and (C). To upper-bound the term (B) we use that the optimal Q-function is Lipschitz with respect to the diminished reward (Proposition 4.3.4):

$$\|\boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(t-1)}}\|_\infty \leq C_z\|\boldsymbol{z}^{(t)} - \boldsymbol{z}^{(t+1)}\|_2, \tag{4.33}$$

where we consider Lipschitzness with respect to the vector $\boldsymbol{z}^\top = [\boldsymbol{w}^\top, \boldsymbol{\lambda}^\top]$. For term (C) we use a very similar property, for fixed policy, Q-values are Lipschitz with respect to the diminished reward (Proposition 4.3.3):

$$\|\boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t-1)}}\|_\infty \leq C_z\|\boldsymbol{z}^{(t)} - \boldsymbol{z}^{(t+1)}\|_2, \tag{4.34}$$

where we consider Lipschitzness w.r.t the vector $\boldsymbol{z}^\top = [\boldsymbol{w}^\top, \boldsymbol{\lambda}^\top]$. This only leaves us with term (A),

$$\|\boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t-1)}} - \boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(t-1)}}\|_\infty \overset{(i)}{=} \|\mathcal{T}_\beta \boldsymbol{Q}^{(t-1)}_{\tilde{\boldsymbol{r}}^{(t-1)}} - \boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(t-1)}}\|_\infty \tag{4.35}$$

$$\overset{(ii)}{=} \|\mathcal{T}_\beta \boldsymbol{Q}^{(t-1)}_{\tilde{r}^{(t-1)}} - \mathcal{T}_\beta \boldsymbol{Q}^{*}_{\tilde{r}^{(t-1)}}\|_\infty \tag{4.36}$$

$$\overset{(iii)}{\leq} \gamma \|\boldsymbol{Q}^{(t-1)}_{\tilde{r}^{(t-1)}} - \boldsymbol{Q}^{*}_{\tilde{r}^{(t-1)}}\|_\infty. \tag{4.37}$$

We use the properties of the soft bellman operator (Definition 2.2.6), specifically in (i) we use that since the policy learning rate η_θ in Algorithm 2 is set to $1/\beta$ the Q-value iterations are given by the soft bellman operator (Proposition 2.4.2), in (ii) we use that the optimal Q is a fixed-point of the soft-bellman optimality operator (Proposition 2.2.2) finally in (iii) we use that the soft-bellman optimality operator is a γ-contraction in the $\|\cdot\|_\infty$ norm (Proposition 2.2.2). Bringing together the bounds (4.32), (4.34) and (4.37) we reach the following upper bound on the Q-value error:

$$\|\boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{*}_{\tilde{r}^{(t)}}\|_\infty \leq \gamma \|\boldsymbol{Q}^{(t-1)}_{\tilde{r}^{(t-1)}} - \boldsymbol{Q}^{*}_{\tilde{r}^{(t-1)}}\|_\infty + \|\boldsymbol{Q}^{*}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{*}_{\tilde{r}^{(t-1)}}\|_\infty + \|\boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{(t)}_{\tilde{r}^{(t-1)}}\|_\infty. \tag{4.38}$$

Now using (4.33) and (4.34) together with the fact that gradient steps in $\boldsymbol{w}$ and $\boldsymbol{\lambda}$ are bounded in norm (by the gradient norm and by the gradient descent step size η_z) we have that:

$$\|\boldsymbol{Q}^{*}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{*}_{\tilde{r}^{(t-1)}}\|_\infty + \|\boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{(t)}_{\tilde{r}^{(t-1)}}\|_\infty \leq 2C_z \|\boldsymbol{z}^{(t)} - \boldsymbol{z}^{(t+1)}\|_2 \tag{4.39}$$

$$\leq 2C_z \eta_z \|\nabla_{\boldsymbol{z}} L(\boldsymbol{\theta}, \dot{\boldsymbol{z}})\|_2 \tag{4.40}$$

$$\leq 2C_z \eta_z (\|\Phi\| + \|\Psi\| + \|\boldsymbol{b}\|_2), \tag{4.41}$$

which is all we need to analyze the local convergence of our algorithm. For conciseness, we let:

$$H^{(t)} := \|\boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{*}_{\tilde{r}^{(t)}}\|_\infty, \tag{4.42}$$

$$C'_z := C_z \eta_z (\|\Phi\| + \|\Psi\| + \|\boldsymbol{b}\|_2). \tag{4.43}$$

Plugging (4.39) into (4.38) (and using $H^{(t)}$ for conciseness) we end up with the following descent inequality:

$$H^{(t)} \leq \gamma H^{(t-1)} + 2C'_z \eta_z, \tag{4.44}$$

unrolling the recursion we get:

$$H^{(t)} \leq \gamma H^{(t-1)} + 2C'_z \eta_z \tag{4.45}$$

$$H^{(t)} \leq \gamma\big(\gamma H^{(t-2)} + 2C'_z \eta_z\big) + 2C'_z \eta_z = \gamma^2 H^{(t-2)} + 2C'_z \eta_z (\gamma + 1) \tag{4.46}$$

$$\vdots$$

$$H^{(t)} \leq \gamma^T H^{(0)} + 2C'_z \eta_z \sum_{t=0}^{T} \gamma^t = \gamma^T H^{(0)} + 2C'_z \eta_z \frac{1-\gamma^T}{1-\gamma}. \tag{4.47}$$

Picking $\eta_z = \frac{1}{T^u}$ (as specified in alg 2) where $u \in (0,1)$ we get the following convergence rate in terms of Q-values:

$$\|\boldsymbol{Q}^{(T)}_{\tilde{r}^{(T)}} - \boldsymbol{Q}^{*}_{\tilde{r}^{(T)}}\|_\infty \leq \|\boldsymbol{Q}^{(0)}_{\tilde{r}^{(0)}} - \boldsymbol{Q}^{*}_{\tilde{r}^{(0)}}\|_\infty \gamma^T + 2C'_z \frac{1-\gamma^T}{1-\gamma} \frac{1}{T^u}. \tag{4.48}$$

Observing that under Assumption 4.2.2:

$$\|\boldsymbol{Q}^{(0)}_{\tilde{r}^{(0)}} - \boldsymbol{Q}^{*}_{\tilde{r}^{(0)}}\|_\infty \leq (1-\gamma) \sum_{t=0}^{+\infty} \gamma^t 2\|\Phi\|_1 = 2\|\Phi\|_1, \tag{4.49}$$

we find:

$$\|\boldsymbol{Q}^{(T)}_{\tilde{r}^{(T)}} - \boldsymbol{Q}^{*}_{\tilde{r}^{(T)}}\|_\infty \leq 2\|\Phi\|_1 \gamma^T + 2C'_z \frac{1-\gamma^T}{1-\gamma} \frac{1}{T^u}. \tag{4.50}$$

Finally, using Proposition 4.3.5 we get a bound of the log-policy error:

$$\|\log \boldsymbol{\pi}^{(T+1)} - \log \boldsymbol{\pi}^{*}_{\tilde{\boldsymbol{r}}^{(T)}}\|_\infty \leq 2\|\boldsymbol{Q}^{(T)}_{\tilde{r}^{(T)}} - \boldsymbol{Q}^{*}_{\tilde{r}^{(T)}}\|_\infty \tag{4.51}$$

$$\leq 4\|\Phi\|_1 \gamma^T + 4C'_z \frac{1-\gamma^T}{1-\gamma} \frac{1}{T^u}. \tag{4.52}$$

Note that $C'_z = 2(\|\Phi\| + \|\Psi\|)(\|\Phi\| + \|\Psi\| + \|\boldsymbol{b}\|_2)$. The proof is complete. □

4.3.4. Global Convergence

Local convergence is established (in Lemma 4.3.3), so we are left with showing global convergence. To this end we will require a way to relate local convergence in the policy to a tight approximation of the dual. Since the quantities over which we optimize can be thought of as scalar products of the occupancy measure and the reward, we formalize this idea by showing that the occupancy measure as a function of the policy is Lipschitz in the $\|\cdot\|_\infty$ norm.

Proposition 4.3.6 (Occupancy is lipschitz w.r.t to the policy in the $\|\cdot\|_\infty$ norm)**.** *We consider any MDP setting, let $\boldsymbol{\mu}$ and $\bar{\boldsymbol{\mu}}$ be two occupancy measures induced by two policies $\boldsymbol{\pi}$ and $\bar{\boldsymbol{\pi}}$, the following upper-bound holds:*

$$\|\boldsymbol{\mu} - \bar{\boldsymbol{\mu}}\|_\infty \leq \frac{1 + \gamma\sqrt{nm}}{1-\gamma}\|\boldsymbol{\pi} - \bar{\boldsymbol{\pi}}\|_\infty. \tag{4.53}$$

The proof is deferred to section B.3.

We are now ready tackle proving the main convergence result: Theorem 4.3.1.

Proof. We allow ourselves to write our Lagrangian as $L(\boldsymbol{\theta}, \boldsymbol{z})$ and our dual as $D(\boldsymbol{z})$. Let us consider the sub-optimality of our function $L(\boldsymbol{\theta}, \boldsymbol{z}) - L(\boldsymbol{\theta}^*, \boldsymbol{z}^*) = L(\boldsymbol{\theta}, \boldsymbol{z}) - L^*$ in terms of the dual and of a perturbation term:

$$L(\boldsymbol{\theta}, \boldsymbol{z}) - L^* = \overbrace{D(\boldsymbol{z}) - L^*}^{(a)} + \overbrace{L(\boldsymbol{\theta}, \boldsymbol{z}) - D(\boldsymbol{z})}^{(b)}. \tag{4.54}$$

So we have a "dual-suboptimality" term (a) and an "dual-approximation error" term (b). To convince ourselves that indeed our dual-descent analysis is sensible and that the approximation error becomes small (i.e. that our algorithm quickly approximates the dual (D)) we will first study term (b). We can bound it using the soft sub-optimality lemma (Lemma 2.4.3) as follows:

$$\begin{aligned}
\left|L(\boldsymbol{\theta}^{(T)}, \boldsymbol{z}^{(T)}) - D(\boldsymbol{z}^{(T)})\right| &\overset{(i)}{=} \Big|\overbrace{J(\boldsymbol{\theta}^{(T)}, \tilde{\boldsymbol{r}}) - J(\boldsymbol{\theta}^E, \tilde{\boldsymbol{r}})}^{=L(\boldsymbol{\theta}^{(T)}, \boldsymbol{z}^{(T)})} - \overbrace{J(\boldsymbol{\theta}^*, \tilde{\boldsymbol{r}}) + J(\boldsymbol{\theta}^E, \tilde{\boldsymbol{r}})}^{=D(\boldsymbol{z}^{(T)})}\Big| && (4.55)\\
&= \left|J(\boldsymbol{\theta}^{(T)}, \tilde{\boldsymbol{r}}) - J(\boldsymbol{\theta}^*, \tilde{\boldsymbol{r}})\right| && (4.56)\\
&\overset{(ii)}{=} \frac{1}{\eta_\theta}\mathbb{E}_{s\sim\boldsymbol{\mu}_s^{\theta^*}}\left[D_{\mathsf{KL}}(\boldsymbol{\pi}^{(t)}(\cdot|s)||\boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}^{(t)}}(\cdot|s))\right] && (4.57)\\
&\leq \frac{1}{\eta_\theta}\max_{s\in S}\left[D_{\mathsf{KL}}(\boldsymbol{\pi}^{(t)}(\cdot|s)||\boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}^{(t)}}(\cdot|s))\right] && (4.58)\\
&\overset{(iii)}{\leq} \frac{1}{\eta_\theta}\max_{s\in S}\left[|\langle \boldsymbol{\pi}^{(t)}_{\tilde{\boldsymbol{r}}}(\cdot|s), \log \boldsymbol{\pi}^{(t)}_{\tilde{\boldsymbol{r}}}(\cdot|s) - \log \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}}(\cdot|s)\rangle|\right] && (4.59)\\
&\overset{(iv)}{\leq} \frac{1}{\eta_\theta}\big[\overbrace{\|\boldsymbol{\pi}^{(t)}_{\tilde{\boldsymbol{r}}}\|_1}^{=1}\cdot\|\log \boldsymbol{\pi}^{(t)}_{\tilde{\boldsymbol{r}}} - \log \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}}\|_\infty\big] && (4.60)\\
&\leq \frac{1}{\eta_\theta}\|\log \boldsymbol{\pi}^{(t)}_{\tilde{\boldsymbol{r}}} - \log \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}}\|_\infty. && (4.61)
\end{aligned}$$

we use strong duality (Proposition 3.2.2) and diminished rewards (Definition 3.2.1). In (i) we just plug in the definitions of L and D, in (ii) we use the soft suboptimality lemma (Lemma 2.4.3). In (iii) we explicitly write out the Kullback-Leibler divergence and then in (iv) we use Hölder's inequality to get to a $\|\cdot\|_\infty$ norm form of our inequality. We can then plug the local optimality result of Lemma 4.3.3 into (4.61) to show that the perturbation terms indeed becomes small as the algorithm progresses:

$$\left|L(\boldsymbol{\theta}^{(T)}, \boldsymbol{z}) - D(\boldsymbol{z})\right| \leq \frac{2}{\eta_\theta}\|\log \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}} - \log \boldsymbol{\pi}^{(t+1)}_{\tilde{\boldsymbol{r}}}\|_\infty \leq \frac{4\|\Phi\|_1}{\eta_\theta}\gamma^T + \frac{4C'_z}{\eta_\theta}\frac{1-\gamma^T}{1-\gamma}\frac{1}{T^u}. \tag{4.62}$$

Note that what happens is that the algorithm converges fast (linearly) to a neighborhood of the optimum, the size of that neighborhood is controlled by the term $\frac{4C'_z}{\eta_\theta}\frac{1-\gamma^T}{1-\gamma}\frac{1}{T^u}$, which can be made arbitrarily small with the choice of an appropriate learning rate $\eta_z = \frac{1}{T^u}$.

Now that we have bounded our "dual-approximation error" term (b) let's move on to studying our "dual-suboptimality" term (a) (recall that we get both of these from equation (4.54)), the term (a) has form:

$$D(\boldsymbol{z}) - L^* = D(\boldsymbol{z}) - D^*, \tag{4.63}$$

since the dual optimum and the minimax optimum coincide. We start our analysis by looking at the gradients used by descent steps of Algorithm 2, i.e. the gradients of the Lagrangian $L(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\lambda})$:

$$\nabla_{\boldsymbol{w}} L(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\lambda}) = \Phi^\top(\boldsymbol{\mu}^{\boldsymbol{\theta}^{(t)}} - \boldsymbol{\mu}^E) = \Phi^\top(\boldsymbol{\mu}^*_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{\mu}^E) + \Phi^\top(\boldsymbol{\mu}^{\boldsymbol{\theta}^{(t)}} - \boldsymbol{\mu}^*_{\tilde{\boldsymbol{r}}^{(t)}}), \tag{4.64}$$

$$\nabla_{\boldsymbol{\lambda}} L(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\lambda}) = \boldsymbol{b} - \Psi^\top \boldsymbol{\mu}^{\boldsymbol{\theta}^{(t)}} = \boldsymbol{b} - \Psi^\top \boldsymbol{\mu}^*_{\tilde{\boldsymbol{r}}^{(t)}} + \Psi^\top(\boldsymbol{\mu}^*_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{\mu}^{\boldsymbol{\theta}^{(t)}}). \tag{4.65}$$

What we are doing here is rearranging the gradients in such a way that we have two components, the gradient of the dual (D1) and a perturbation term:

$$\nabla_{\boldsymbol{w}} L(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\lambda}) = \nabla_{\boldsymbol{w}} D(\boldsymbol{w}, \boldsymbol{\lambda}) + \boldsymbol{\sigma}^{(t)}_{\boldsymbol{w}}, \tag{4.66}$$

$$\nabla_{\boldsymbol{\lambda}} L(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\lambda}) = \nabla_{\boldsymbol{w}} D(\boldsymbol{w}, \boldsymbol{\lambda}) + \boldsymbol{\sigma}^{(t)}_{\boldsymbol{\lambda}}. \tag{4.67}$$

We write the gradients used by our gradient descent algorithm at iteration t as $\boldsymbol{g}^{(t)}$, specifically, we have:

$$\boldsymbol{g}^{(t)} = \begin{bmatrix} \nabla_{\boldsymbol{w}} D(\boldsymbol{w}, \boldsymbol{\lambda}) \\ \nabla_{\boldsymbol{w}} D(\boldsymbol{w}, \boldsymbol{\lambda}) \end{bmatrix} + \begin{bmatrix} \boldsymbol{\sigma}^{(t)}_{\boldsymbol{w}} \\ \boldsymbol{\sigma}^{(t)}_{\boldsymbol{\lambda}} \end{bmatrix} = \nabla_{\boldsymbol{z}} D(\boldsymbol{z}^{(t)}) + \boldsymbol{\sigma}^{(t)}_{\boldsymbol{z}}. \tag{4.68}$$

Using dual convexity (Proposition 4.3.2) and strong-duality (Proposition 3.2.2) we have that:

$$D(\boldsymbol{z}^{(t)}) - D^* = D(\boldsymbol{z}^{(t)}) - L^* \leq \langle \nabla_{\boldsymbol{z}} D(\boldsymbol{z}^{(t)}), \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle \tag{4.69}$$

$$\overset{(i)}{=} \langle \boldsymbol{g}^{(t)} - \boldsymbol{\sigma}^{(t)}_{\boldsymbol{z}}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle \tag{4.70}$$

$$= \langle \boldsymbol{g}^{(t)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle - \langle \boldsymbol{\sigma}^{(t)}_{\boldsymbol{z}}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle, \tag{4.71}$$

where (i) holds by plugging (4.68) into the gradient expression. We thus have two terms, a main gradient step term $\langle \boldsymbol{g}^{(t)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle$ which describes the iterations of our algorithm and a perturbation term $\langle \boldsymbol{\sigma}_{\boldsymbol{z}^{(t)}}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle$. First, we will show that the perturbations are sufficiently small, since we do not know the sign of the scalar product we will bound it in absolute values:

$$|\langle \boldsymbol{\sigma}^{(t)}_{\boldsymbol{z}}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle| \overset{(i)}{\leq} \|\boldsymbol{\sigma}^{(t)}_{\boldsymbol{z}}\|_\infty \cdot \|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_1 \tag{4.72}$$

$$\overset{(ii)}{\leq} D_{1,z} \|\boldsymbol{\sigma}^{(t)}_{\boldsymbol{z}}\|_\infty. \tag{4.73}$$

Where we just use Hölder's inequality in (i) and in (ii) plug in the diameter $D_{1,z}$ of the domain of $\boldsymbol{z}$ (in the $\|\cdot\|_1$ norm). Let us now concentrate on upper bounding the $\|\cdot\|_\infty$ norm of the perturbation term $\boldsymbol{\sigma}^{(t)}_{\boldsymbol{z}}$:

$$\|\boldsymbol{\sigma}^{(t)}_{\boldsymbol{z}}\|_\infty = \left\| \begin{bmatrix} \boldsymbol{\sigma}_{\boldsymbol{w}} \\ \boldsymbol{\sigma}_{\boldsymbol{\lambda}} \end{bmatrix} \right\|_\infty = \left\| \begin{bmatrix} \Phi^\top \\ \Psi^\top \end{bmatrix} (\boldsymbol{\mu}^{\boldsymbol{\theta}^{(t)}} - \boldsymbol{\mu}^*_{\tilde{\boldsymbol{r}}}) \right\|_\infty \tag{4.74}$$

$$\overset{(i)}{\leq} \left\| \begin{bmatrix} \Phi^\top \\ \Psi^\top \end{bmatrix} \right\|_\infty \left\| \boldsymbol{\mu}^{\boldsymbol{\theta}^{(t)}} - \boldsymbol{\mu}^*_{\tilde{\boldsymbol{r}}} \right\|_\infty \leq \left(\|\Phi^\top\|_\infty + \|\Psi^\top\|_\infty \right) \left\| \boldsymbol{\mu}^{\boldsymbol{\theta}^{(t)}} - \boldsymbol{\mu}^*_{\tilde{\boldsymbol{r}}} \right\|_\infty \tag{4.75}$$

$$\overset{(ii)}{\leq} \sqrt{k+d} \left(\|\Phi\| + \|\Psi\| \right) \left\| \boldsymbol{\mu}^{\boldsymbol{\theta}^{(t)}} - \boldsymbol{\mu}^*_{\tilde{\boldsymbol{r}}} \right\|_\infty = \frac{C_z \sqrt{k+d}}{2} \left\| \boldsymbol{\mu}^{\boldsymbol{\theta}^{(t)}} - \boldsymbol{\mu}^*_{\tilde{\boldsymbol{r}}} \right\|_\infty \tag{4.76}$$

$$\overset{(iii)}{\leq} \frac{C_z \sqrt{k+d}(1+\gamma\sqrt{nm})}{2(1-\gamma)} \left\| \boldsymbol{\pi}_{\boldsymbol{\theta}^{(t)}} - \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}} \right\|_\infty \tag{4.77}$$

$$\overset{(iv)}{\leq} \frac{C_z \sqrt{k+d}(1+\gamma\sqrt{nm})}{2(1-\gamma)} \left\| \log \boldsymbol{\pi}_{\boldsymbol{\theta}^{(t)}} - \log \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}} \right\|_\infty. \tag{4.78}$$

Where in (i) we take the maximum row sum norm on our matrix (which we write $\|\cdot\|_\infty$ as it is the operator norm associated with the $\|\cdot\|_\infty$ norm on vectors). In (ii) we upper-bound the maximum row sum norm by with the spectral norm, in (iii) we make use of Proposition 4.3.6 (lipschitzness of the occupancy measure as a function of the policy in the $\|\cdot\|_\infty$ norm) to bound the distance in occupancy measure by the distance in policy. Finally, in (iv) we use that on the relevant domain ($\left\|\boldsymbol{\pi}_{\boldsymbol{\theta}^{(t)}} - \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}}\right\|_\infty < 1$) the $\|\cdot\|_\infty$ norm of the difference of the logs upper bounds the $\|\cdot\|_\infty$ norm of the difference. Putting together (4.78), (4.73) and using Lemma 4.3.3 we show:

$$|\langle \boldsymbol{\sigma}^{(t)}_{\boldsymbol{z}}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^*\rangle| \leq \frac{C_z\sqrt{k+d}(1+\gamma\sqrt{nm})D_{1,z}}{2(1-\gamma)}\big(4\|\Phi\|_1\gamma^t + 4C'_z\frac{1-\gamma^t}{1-\gamma}\frac{1}{t^u}\big). \tag{4.79}$$

Now confident that gradient perturbations indeed decrease fast with iterations of our algorith we get back to (4.71) we now consider the gradient step term $\langle \boldsymbol{g}^{(t)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^*\rangle$ note that the steps taken by gradient descent are given by $\boldsymbol{g}^{(t)} = \frac{1}{\eta_\theta}\boldsymbol{z}^{(t)} - \boldsymbol{z}^{(t+1/2)}$:

$$\begin{aligned}
\langle \boldsymbol{g}^{(t)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^*\rangle &= \frac{1}{\eta_z}\langle \boldsymbol{z}^{(t)} - \boldsymbol{z}^{(t+1/2)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^*\rangle && (4.80)\\
&\overset{(i)}{=} \frac{1}{2\eta_z}\Big(\|\boldsymbol{z}^{(t+1/2)} - \boldsymbol{z}^{(t)}\|_2^2 + \|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2^2 - \|\boldsymbol{z}^{(t+1/2)} - \boldsymbol{z}^*\|_2^2\Big) && (4.81)\\
&\overset{(ii)}{\leq} \frac{1}{2\eta_z}\Big(\eta_z^2\|\boldsymbol{g}^{(t)}\|_2^2 + \|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2^2 - \|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^*\|_2^2\Big). && (4.82)
\end{aligned}$$

Where (i) uses the parallelogram law and (ii) the non-expansiveness of the projection operator. Plugging the bounds (4.82) and (4.79) into the descent inequality (4.71) we have:

$$\begin{aligned}
D(\boldsymbol{z}^{(t)}) - L^* &\leq \frac{1}{2\eta_z}\Big(\eta_z^2\|\boldsymbol{g}^{(t)}\|_2^2 + \|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2^2 - \|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^*\|_2^2\Big) && (4.83)\\
&\quad + \frac{C_z\sqrt{k+d}(1+\gamma\sqrt{nm})D_{1,z}}{2(1-\gamma)}\big(4\|\Phi\|_1\gamma^T + 4C'_z\frac{1-\gamma^T}{1-\gamma}\frac{1}{T^u}\big).
\end{aligned}$$

For readability we let $C_{\mathsf{p}} = \frac{C_z\sqrt{k+d}(1+\gamma\sqrt{nm})D_{1,z}}{2(1-\gamma)}$. Taking empirical means on both sides we get:

$$\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T-1}(D(\boldsymbol{z}^{(t)}) - L^*) &\leq \frac{1}{2\eta_z T}\sum_{t=0}^{T-1}\Big(\eta_z^2\|\boldsymbol{g}^{(t)}\|_2^2 + \|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2^2 - \|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^*\|_2^2\Big) && (4.84)\\
&\quad + \frac{C_{\mathsf{p}}}{T}\left(4\|\Phi\|_1\sum_{t=0}^{T-1}\gamma^t + 4C'_z\sum_{t=0}^{T-1}\frac{1-\gamma^t}{1-\gamma}\frac{1}{T^u}\right)\\
&\leq \frac{1}{2\eta_z T}\sum_{t=0}^{T-1}\Big(\eta_z^2\|\boldsymbol{g}^{(t)}\|_2^2 + \|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2^2 - \|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^*\|_2^2\Big) && (4.85)\\
&\quad + \frac{4\|\Phi\|_1 C_{\mathsf{p}}}{T}\frac{1-\gamma^T}{1-\gamma} + \frac{4C'_z C_{\mathsf{p}}}{T^u}\\
&= \frac{1}{2\eta_z T}\sum_{t=0}^{T-1}\Big(\eta_z^2\|\boldsymbol{g}^{(t)}\|_2^2\Big) + \frac{\|\boldsymbol{z}^{(0)} - \boldsymbol{z}^*\|_2^2 - \|\boldsymbol{z}^{(T)} - \boldsymbol{z}^*\|_2^2}{2\eta_z T} && (4.86)\\
&\quad + \frac{4\|\Phi\|_1 C_{\mathsf{p}}}{T}\frac{1-\gamma^T}{1-\gamma} + \frac{4C'_z C_{\mathsf{p}}}{T^u}\\
&\leq \frac{(\|\Psi\|^2 + \|\Phi\|^2 + \|\boldsymbol{b}\|_2^2)}{2T^u} + \frac{\|\boldsymbol{z}^{(0)} - \boldsymbol{z}^*\|_2^2 T^u}{2T} && (4.87)\\
&\quad + \frac{4\|\Phi\|_1 C_{\mathsf{p}}}{T}\frac{1-\gamma^T}{1-\gamma} + \frac{4C'_z C_{\mathsf{p}}}{T^u}.
\end{aligned}$$

Rearranging, and introducing back the upper-bound (4.62) on the perturbation term (b) ($L(\boldsymbol{z}^{(t)}, \boldsymbol{\theta})^{(t)} - L^* = D(\boldsymbol{z}^{(t)}) - L^* + L(\boldsymbol{z}^{(t)}, \boldsymbol{\theta})^{(t)} - D(\boldsymbol{z}^{(t)})$) we have:

$$\frac{1}{T}\sum_{t=0}^{T-1}(L(\boldsymbol{z}^{(t)}) - L^*) \leq \left(\frac{(\|\Psi\|^2 + \|\Phi\|^2 + \|\boldsymbol{b}\|_2^2)}{2} + 4C'_z C_{\mathsf{p}} + 4\beta C'_z\frac{1-\gamma^T}{1-\gamma}\right)T^{-u} + \frac{\|\boldsymbol{z}^{(0)} - \boldsymbol{z}^*\|_2^2}{2}T^{u-1} \tag{4.88}$$

$$+ \frac{4\|\Phi\|_1 C_{\mathsf{p}}}{T} \frac{1-\gamma^T}{1-\gamma} + 4\beta\|\Phi\|_1 \gamma^T.$$

convergence is fastest when we pick u^*:

$$u^* = \arg\min_u \{-u, u-1\}, \tag{4.89}$$

i.e. the optimal learning rate is $\eta_z = \frac{1}{\sqrt{T}}$, explicitly introducing both learning rates ($\eta_\theta = 1/\beta$) we have the following convergence rate:

$$\frac{1}{T}\sum_{t=0}^{T-1}(L(\boldsymbol{\theta}^{(t)}, \boldsymbol{z}^{(t)}) - L^*) \leq \left(\frac{(\|\Psi\|^2 + \|\Phi\|^2 + \|\boldsymbol{b}\|_2^2 + \|\boldsymbol{z}^{(0)} - \boldsymbol{z}^*\|_2^2)}{2} + 4C_z' C_{\mathsf{p}} + 4\beta C_z' \frac{1-\gamma^T}{1-\gamma} \right) \frac{1}{\sqrt{T}} \tag{4.90}$$

$$\begin{aligned} &+ \frac{4\|\Phi\|_1 C_{\mathsf{p}}}{T} \frac{1-\gamma^T}{1-\gamma} + 4\beta\|\Phi\|_1 \gamma^T \\ &= C_G \frac{1}{\sqrt{T}} + C_{\mathsf{pert}} \frac{1-\gamma^T}{1-\gamma} \frac{1}{T}. + 4\beta\|\Phi\|_1 \gamma^T. \end{aligned} \tag{4.91}$$

We complete the proof by observing that since the minimum element in the sequence $(D(\boldsymbol{z}^{(t)}) - L^*)$ over which we take the empirical mean, there must be some iteration (t) for which the left-hand side upper-bounds the suboptimality. □

5

Convergence with stochastic gradients, sample complexity

In the following chapter, we discuss and analyze *NPG-CIRL* using Monte-Carlo sampling to estimate the gradients.

5.1. The algorithm, gradient estimators

Recall that in order to solve *CIRL*, *NPG-CIRL* (Algorithm 2) runs T simultaneous gradient descent-ascent iterations of the form:

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} + \eta_\theta (\mathfrak{F}^{\boldsymbol{\theta}})^\dagger \nabla_{\boldsymbol{\theta}} L(\boldsymbol{w}^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{\theta}^{(t)}), \tag{5.1}$$

$$\boldsymbol{w}^{(t+1)} \leftarrow \mathcal{P}_{\mathsf{dom}(\boldsymbol{w})}\big(\boldsymbol{w}^{(t)} - \eta_z \nabla_{\boldsymbol{w}} L(\boldsymbol{w}^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{\theta}^{(t)})\big), \tag{5.2}$$

$$\boldsymbol{\lambda}^{(t+1)} \leftarrow \mathcal{P}_{\boldsymbol{\lambda}\in\Lambda}\big(\boldsymbol{\lambda}^{(t)} - \eta_z \nabla_{\boldsymbol{\lambda}} L(\boldsymbol{w}^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{\theta}^{(t)})\big). \tag{5.3}$$

In Chapter 4, we assumed oracle access to exact gradients $\nabla_{\boldsymbol{\theta}} L$, $\nabla_{\boldsymbol{w}} L$ and $\nabla_{\boldsymbol{\lambda}} L$. Such access is, however, only possible when the dynamics of the MDP are exactly known. In order to make our algorithm practical, we suggest a stochastic implementation in which gradients are estimated in a Monte-Carlo fashion from a batch of samples. We denote our gradient estimators as:

$$\boldsymbol{g}_{\boldsymbol{\theta}}^{(t)} = \hat{\nabla}_{\boldsymbol{\theta}} L(\boldsymbol{w}^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{\theta}^{(t)}), \tag{5.4}$$

$$\boldsymbol{g}_{\boldsymbol{w}}^{(t)} = \hat{\nabla}_{\boldsymbol{w}} L(\boldsymbol{w}^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{\theta}^{(t)}), \tag{5.5}$$

$$\boldsymbol{g}_{\boldsymbol{\lambda}}^{(t)} = \hat{\nabla}_{\boldsymbol{\lambda}} L(\boldsymbol{w}^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{\theta}^{(t)}). \tag{5.6}$$

In the analysis of Chapter 4, we also have assumed direct access to the expert policy (equivalently, to its occupancy measure). This is the setting of Problem 3.1.1. Again this assumption is not reasonable. When learning from a real dataset, the expert policy can only be accessed through the expert dataset $\mathcal{D}$, which consists in a set of $B_{\mathcal{D}}$ truncated trajectories of length H:

$$\mathcal{D}^{(t)} := \big\{\{s_i^{(h)}, a_i^{(h)}, r_i^{(h)}, \boldsymbol{\Psi}_i^{(h)}\}_{h=0}^{H-1}\big\}_{i=1}^{B_{\mathcal{D}}}. \tag{5.7}$$

This is the setting of Problem 3.1.2. Estimating the expert policy by sampling $\mathcal{D}$ introduces an imprecision. We will thus also discuss how the number of samples in the dataset $\mathcal{D}$, affects the quality of the recovered solution. Although one could, with enough samples, compute an arbitrarily good estimator of $\boldsymbol{\pi}^E \in \Delta_A^S \subset \mathbb{R}^{nm}$ (or equivalently of $\boldsymbol{\mu}^E \in \mathcal{M} \subset \mathbb{R}^{nm}$), the number of samples might become very high when nm increases. For that reason, when considering linear reward classes that allow for parameterizing the reward with a lower dimensional vector $\boldsymbol{w} \in \mathbb{R}^k$ ($k < nm$), estimating the $\boldsymbol{\pi}^E$ or $\boldsymbol{\mu}^E$ vector is inefficient[1]. This motivates the introduction of the *feature expectation*.

[1] It is inefficient in the sense that it involves estimating a vector $\boldsymbol{\mu} \in \mathbb{R}^{nm}$ in order to later multiply it with the reward feature matrix Φ to compute a lower-dimensionality gradient $\boldsymbol{g}_{\boldsymbol{w}} \in \mathbb{R}^k$.

Definition 5.1.1 (Feature expectation)**.** *Consider some linear reward class $\mathcal{R}_\Phi$ with feature matrix Φ, and some policy π that induces an occupancy measure $\boldsymbol{\mu}^{\boldsymbol{\pi}}$. We call the vector:*

$$\boldsymbol{\varphi}^{\boldsymbol{\pi}} = \Phi^\top \boldsymbol{\mu}^{\boldsymbol{\pi}}, \tag{5.8}$$

the feature expectation associated with the policy π.

Note that computing the feature expectation of a policy provides a way of computing the return as follows:

$$J(\boldsymbol{\theta}, \boldsymbol{w}) = \mathbb{E}_{\tau \sim \boldsymbol{\pi}_{\boldsymbol{\theta}}}\big[\sum_{t=0}^{\infty} \gamma^t r_{\boldsymbol{w}}(s_t, a_t)\big] = \langle \boldsymbol{r}_{\boldsymbol{w}}, \boldsymbol{\mu}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}} \rangle \tag{5.9}$$

$$= \langle \Phi \boldsymbol{w}, \boldsymbol{\mu}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}} \rangle = \langle \boldsymbol{w}, \Phi^\top \boldsymbol{\mu}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}} \rangle \tag{5.10}$$

$$= \langle \boldsymbol{w}, \boldsymbol{\varphi}^{\boldsymbol{\theta}} \rangle. \tag{5.11}$$

Similarly, the feature expectation appears in the computation of the gradient $\nabla_{\boldsymbol{w}} L(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\lambda})$, we have:

$$\nabla_{\boldsymbol{w}} L(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\lambda}) = \Phi^\top (\boldsymbol{\mu}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}} - \boldsymbol{\mu}^E), \tag{5.12}$$

$$= \boldsymbol{\varphi}^{\boldsymbol{\theta}} - \boldsymbol{\varphi}^E. \tag{5.13}$$

Practically our stochastic implementation of *NPG-CIRL* takes the form of Algorithm 3. It starts by running an estimation subroutine on the expert dataset $\mathcal{D}$ and computes an estimate $\hat{\boldsymbol{\varphi}}^E$ of $\boldsymbol{\varphi}^E$, the feature expectation of the expert policy $\boldsymbol{\pi}^E$. Then, Algorithm 3 runs iterations analogue to the ones of Algorithm 2, except instead of directly accessing gradient oracles, it samples a batch

$$\mathcal{B}^{(t)} := \big\{\{s_i^{(h)}, a_i^{(h)}, r_i^{(h)}, \boldsymbol{\Psi}_i^{(h)}\}_{h=0}^{H-1}\big\}_{i=1}^{B} \tag{5.14}$$

of trajectories generated with the current policy $\boldsymbol{\pi}^{(t)}$ and uses it to compute gradient estimates $\boldsymbol{g}_{\boldsymbol{\theta}}^{(t)}, \boldsymbol{g}_{\boldsymbol{w}}^{(t)}, \boldsymbol{g}_{\boldsymbol{\lambda}}^{(t)}$ via Monte-Carlo sampling.

Algorithm 3: NPG-CIRL (Sampled Gradients)

Set the learning rates $0 < \eta_\theta < \frac{1}{\beta}$, $\eta_z = \frac{1}{\sqrt{T}}$
Estimate the feature expectation $\hat{\boldsymbol{\varphi}}^E$ from the expert dataset D^E.
Initialize the algorithm at some point $(\boldsymbol{\theta}^{(0)}, \boldsymbol{w}^{(0)}, \boldsymbol{\lambda}^{(0)})$
for *iteration* $t = 1, 2, ..., T$ **do**
  Sample a batch of trajectories.
  for *batch* $i = 1, 2, ..., B$ **do**
    draw $s_0^{(i)} \sim \boldsymbol{\nu}_0$
    for *step* $h = 1, ...H$ **do**
      pick $a_i^{(h-1)} \sim \boldsymbol{\pi}^{(t)}(\cdot | s_i^{(h-1)}, \boldsymbol{\theta}_t)$
      draw $s_i^{(h)} \sim P(s_i^{(h)} | s_i^{(h-1)}, a_i^{(h-1)})$
      $r_h^{(i)} \leftarrow r(s_i^{(h)}, a_i^{(h)})$
      $\boldsymbol{\Psi}_h^{(i)} \leftarrow \boldsymbol{\Psi}(s_i^{(h)}, a_i^{(h)})$
    end
  end
  Compute the gradient estimates $\boldsymbol{g}_{\boldsymbol{\theta}}^{(t)}, \boldsymbol{g}_{\boldsymbol{w}}^{(t)}, \boldsymbol{g}_{\boldsymbol{\lambda}}^{(t)}$ using the trajectories
  $\big\{\{s_i^{(h)}, a_i^{(h)}, r_i^{(h)}, \boldsymbol{\Psi}_h^{(i)}\}_{h=0}^{H-1}\big\}_{i=1}^{B}$.
  $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} + \eta_\theta (\tilde{\boldsymbol{\mathfrak{F}}}^{\boldsymbol{\theta}})^\dagger \hat{\boldsymbol{g}}_\theta$
  $\boldsymbol{w}^{(t+1)} \leftarrow \mathcal{P}_{\mathsf{dom}(\boldsymbol{w})}\big(\boldsymbol{w}^{(t)} - \eta_z \hat{\boldsymbol{g}}_w\big)$
  $\boldsymbol{\lambda}^{(t+1)} \leftarrow \mathcal{P}_{\boldsymbol{\lambda} \in \Lambda}\big(\boldsymbol{\lambda}^{(t)} - \eta_z \hat{\boldsymbol{g}}_\lambda\big)$
end
return $(\boldsymbol{w}^{(T)}, \boldsymbol{\lambda}^{(T)}, \boldsymbol{\theta}^{(T)})$.

Algorithm 3 heavily depends on the choice of the gradient estimators. In the following subsection (SubSection 5.2.2) we propose estimators for $\boldsymbol{g}_{\boldsymbol{\theta}}$, $\boldsymbol{g}_{\boldsymbol{w}}$ and $\boldsymbol{g}_{\boldsymbol{\lambda}}$. We draw the reader's attention to the fact that the choice of gradient estimators we suggest is not the only option. It might be relevant

when implementing to consider different approaches[2]. We use a trajectory-based Monte Carlo estimator for the policy-gradient estimation.

Assumption 5.1.1 (GPOMDP estimator)**.** *We estimate policy gradients using the GPOMDP estimator[3] [Baxter and Bartlett 2001], which approaches computing the gradient as:*

$$\boldsymbol{g}_{\theta}^{(t)} = \frac{1}{B}\sum_{i=1}^{B}\left(\sum_{h=0}^{H-1}\gamma^{h}\left(\sum_{j=0}^{h}\nabla_{\theta}\log\pi_{\boldsymbol{\theta}^{(t)}}(a_j^i|s_j^i)\right)(\tilde{r}_{\boldsymbol{\lambda}^{(t)}} - \beta\log\pi_{\boldsymbol{\theta}^{(t)}})\right). \tag{5.15}$$

As we already hinted with equation (5.13), the computation of the reward gradient will rely on the feature expectation. We thus first describe a generic estimator for the feature expectation (which we will both use for reward gradients and for estimating the feature expectation of the expert policy from the dataset $\mathcal{D}$). That estimator is the following:

$$\hat{\boldsymbol{\varphi}} = \frac{1-\gamma}{B}\sum_{i=1}^{B}\Big(\sum_{h=0}^{H-1}\gamma^{h}\boldsymbol{\Phi}(s_i^{(h)}, a_i^{(h)})\Big), \tag{5.16}$$

where $\boldsymbol{\Phi}(s_i^{(h)}, a_i^{(h)}) \in \mathbb{R}^f$ is the column of the feature matrix Φ associated with the state-action pair $(s_i^{(h)}, a_i^{(h)})$.

Assumption 5.1.2 (Reward gradient estimator)**.** *We use estimator (5.16) to compute the reward gradient $\boldsymbol{g}_{\boldsymbol{w}}^{(t)}$ as follows:*

$$\boldsymbol{g}_{\boldsymbol{w}}^{(t)} = \hat{\boldsymbol{\varphi}}^{\boldsymbol{\theta}} - \hat{\boldsymbol{\varphi}}^{E}, \tag{5.17}$$

where $\hat{\boldsymbol{\varphi}}^{\boldsymbol{\theta}}$ is computed from the last batch $\mathcal{B}^{(t)}$ while $\hat{\boldsymbol{\varphi}}^{E}$ is computed only once using the expert dataset $\mathcal{D}$.

Finally, we discuss an estimator for the Lagrangian multiplier gradients $\boldsymbol{g}_{\boldsymbol{\lambda}}^{(t)}$.

Assumption 5.1.3 (Lagrange multiplier gradient estimator)**.** *We estimate the Lagrange multiplier gradient by:*

$$\boldsymbol{g}_{\boldsymbol{\lambda}}^{(t)} = \boldsymbol{b} - \frac{1-\gamma}{B}\sum_{i=1}^{B}\Big(\sum_{h=0}^{H-1}\gamma^{h}\boldsymbol{\Psi}(s_i^{(h)}, a_i^{(h)})\Big), \tag{5.18}$$

where $\boldsymbol{\Psi}(s,a) \in \mathbb{R}^d$ is the cost encountered in state-action pair (s,a).

We note that all three gradient estimators that we previously defined are biased. We discuss this in more detail later, but it is a key design choice of our algorithm to choose to use biased gradient estimators. It would alternatively be possible to work with unbiased gradient estimators (using geometric sampling, as in [Y. Ding, Zhang, and Lavaei 2021], Section 3.2). We leverage the fact that our gradient descent method is robust to small biases in gradients and use biased gradient estimators. This presents the advantage of greatly simplifying implementation.

5.2. Analysis

5.2.1. Analysis sketch, big picture

Our analysis builds upon two main pillars; first (Section 5.2.2), we discuss the properties of the estimators. We show that the mean Euclidian distance error (*MEDE*) of the estimators that we use is non-zero but is bounded and can be scaled down arbitrarily with the number of samples. Once our estimators are characterized, we abstract them away and analyze the convergence of Algorithm 3

[2] One such technique which we do not analyze but which might actually be quite efficient in practice is an actor-critic approach to gradient computation for the policy.

[3] The GPOMDP estimator inherits its name from the algorithm for which it was originally derived [Baxter and Bartlett 2001], which described a policy algorithm for partially observable MDPs. The acronym GPOMDP stands for "gradient of a partially observable Markov decision process". It has since become a very commonly used estimator for policy gradient, including in non-partially observable settings.

with stochastic oracles for the gradient and for the expert policy. Similarly, as what we did in Chapter 4, we first show fast convergence to a neighbourhood of the locally optimal policy (Section 5.2.4), and then, we consider global convergence (Section 5.2.5). Finally equipped with our convergence results as well as with well-established properties for our gradient and expert policy estimators and compute bounds for sample complexity (Section 5.2.6).

5.2.2. Properties of the estimators

We now analyze the properties of all three gradient estimators as well as of the feature expectation estimator. We focus on characterizing *mean euclidean distance error* (*MEDE*) and start with the *GPOMDP* estimator.

Proposition 5.2.1 (GPOMDP has bounded *MEDE*)**.** *The gradient estimator $\boldsymbol{g}_{\boldsymbol{\theta}}^{(t)}$ as defined in (5.15) satisfies:*

$$\mathbb{E}\Big[\|\boldsymbol{g}_{\theta}^{(t)} - \nabla_{\theta} L(\boldsymbol{\theta}^{(t)}, \boldsymbol{w}^{(t)}, \boldsymbol{\lambda}^{(t)})\|_2\Big] \leq \sqrt{2} b_{max} \frac{\gamma^2(2-\gamma)}{(1-\gamma)^2} H\gamma^{H-1} + \frac{2\sqrt{6}\sqrt{\|\Phi\|_1 + \lambda_{max} + 24\beta^2(\log m)^2}}{B(1-\gamma)^2}, \tag{5.19}$$

where $b_{max} := \|\Phi\| + \lambda_{max} + \beta \log \pi_{min}$. The proof is deferred to appendix C.2.1.

Next, we characterize the properties of the reward and Lagrangian multiplier gradient estimator.

Proposition 5.2.2 (Reward gradient estimator has bounded *MEDE*)**.** *The gradient estimator $\boldsymbol{g}_{\boldsymbol{w}}^{(t)}$ as defined in (5.17) satisfies:*

$$\mathbb{E}\Big[\|\boldsymbol{g}_{\boldsymbol{w}}^{(t)} - \nabla_{\boldsymbol{w}} L(\boldsymbol{\theta}^{(t)}, \boldsymbol{w}^{(t)}, \boldsymbol{\lambda}^{(t)})\|_2\Big] \leq \Phi_{max}\left(\frac{2\sqrt{1-\gamma}}{B} + \gamma^H\right), \tag{5.20}$$

where Φ_{max} denotes the maximum $\|\cdot\|_2$ norm of any column of the constraint matrix Ψ. The proof is deferred to appendix C.2.2.

Lastly, we study the error of the Lagrangian multiplier gradient estimator, which has the exact same form as the feature expectation gradient estimator. Thus, we omit the derivation and simply state the result.

Proposition 5.2.3 (The Lagrangian gradient estimator)**.** *The gradient estimator $\boldsymbol{g}_{\boldsymbol{w}}^{(t)}$ as defined in (5.18) satisfies:*

$$\mathbb{E}\Big[\|\boldsymbol{g}_{\boldsymbol{\lambda}}^{(t)} - \nabla_{\boldsymbol{\lambda}} L(\boldsymbol{\theta}^{(t)}, \boldsymbol{w}^{(t)}, \boldsymbol{\lambda}^{(t)})\|_2\Big] \leq \Psi_{max}\left(\frac{2\sqrt{1-\gamma}}{B} + \gamma^H\right), \tag{5.21}$$

where Ψ_{max} denotes the maximum $\|\cdot\|_2$ norm of any column of the constraint matrix Ψ.

5.2.3. Setting for the stochastic convergence analysis

We formally state the additional assumptions that are required for proving the convergence in the stochastic setting. As we did in our exact gradient analysis, we will use the optimization variable $\boldsymbol{z}^{(t}$ that concatenates the dual variables $\boldsymbol{w}^{(t)}$ and $\boldsymbol{\lambda}^{(t)}$:

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{\lambda} \end{bmatrix}. \tag{5.22}$$

Whenever taking expectations over a step of the stochastic gradient algorithms, we will make use of the notation $\mathbb{E}_{(t+1)}$ to denote that the expectation is taken over the randomness induced by the stochastic gradient estimates from iteration (t). To keep our analysis clearer, we will - in this subsection and the next one - abstract away the gradient estimators and rather consider bounded-bias gradient oracles.

Assumption 5.2.1 (Bounded PG MEDE). *We assume that the mean Euclidian distance error (MEDE) in the policy-gradient estimators is bounded:*

$$\mathbb{E}\Big[\|\boldsymbol{g}_{\boldsymbol{\theta}} - \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}^{(t)}, \boldsymbol{w}^{(t)}, \boldsymbol{\lambda}^{(t)})\|_\infty\Big] \leq \delta. \tag{5.23}$$

Assumption 5.2.2 (Bounded z-MEDE). *We assume that the mean Euclidian distance error MEDE in the z gradient estimators is bounded:*

$$\mathbb{E}\Big[\|\boldsymbol{g}_{\boldsymbol{z}} - \nabla_{\boldsymbol{z}} L(\boldsymbol{\theta}^{(t)}, \boldsymbol{w}^{(t)}, \boldsymbol{\lambda}^{(t)})\|_\infty\Big] \leq \delta_z. \tag{5.24}$$

The result our analysis builds towards takes the form of Theorem 5.2.1, which relates average Lagrangian suboptimality of Algorithm 3 to the number of iterations and to the quality of the gradient estimates (which is measured in terms of Assumptions 5.2.1 and 5.2.2).

Theorem 5.2.1 (Global convergence of stochastic NPG-CIRL). *Consider the sequence of policy parameters $\{\boldsymbol{\theta}^{(t)}\}_{t=1}^T$, of reward parameters $\{\boldsymbol{w}^{(t)}\}$ and of Lagrangian multipliers $\{\boldsymbol{\lambda}^{(t)}\}$ generated by running T steps of Algorithm 3 with policy-learning rate $\eta_\theta = 1/\beta$, learning rate $\eta_z = \frac{1}{\sqrt{T}}$. Furthermore, assume Assumptions 4.2.1 (softmax policy), 4.2.2 (linear reward class), 4.2.3 (negative Shanon regularizer), 4.2.4 (non-vanishing occupancy measure), 5.2.1 (policy gradient oracle) and 5.2.2 (z-gradient oracle) are satisfied. The average Lagrangian suboptimality attained by the sequence satisfies:*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} L(\boldsymbol{z}^{(t)}, \boldsymbol{\theta}^{(t)})\right] - L^* = O\Big(\frac{1}{\sqrt{T}} + \delta + \delta_z\Big). \tag{5.25}$$

Where δ is the upper bound on the MEDE of the GPOMDP estimator (Assumption 5.2.1) and δ_z is the upper bound on the MEDE of the z-gradient estimator (Assumption 5.2.2).

We now proceed with the convergence analysis of the algorithm. We first provide a result equivalent to Lemma 4.3.3, namely a local optimality lemma, and then we move on to global convergence (in the next subsection). What we will show is that Algorithm 3 converges *in expectation*.

5.2.4. Convergence to a local optimum

Lemma 5.2.2 (Bounded Q-value bias in the policy update expression). *Under Assumption 5.2.1 (policy gradient oracle) and 4.2.4 (non-vanishing occupancy measure), we have policy updates that for any $(s,a) \in S \times A$ satisfy:*

$$\mathbb{E}_{(t+1)}[\pi^{(t+1)}(a|s)] = \big(\pi^{(t)}(a|s)\big)^{1-\eta_\theta\beta} \exp\big(\eta_\theta \hat{Q}_{\tilde{\boldsymbol{r}}}^{(t)}(s,a)\big), \tag{5.26}$$

where $\hat{Q}_r^{(t)}$ satisfies:

$$\big\|\hat{\boldsymbol{Q}}_{\tilde{\boldsymbol{r}}}^{(t)} - \boldsymbol{Q}_{\tilde{\boldsymbol{r}}}^{(t)}\big\|_\infty \leq \frac{\delta}{\mu_{min}} = \breve{\delta}. \tag{5.27}$$

Where $\mu_{min} = \min_{s\in S} \sum_{a\in A} \mu(s,a)$. The proof is deferred to appendix C.1

Lemma 5.2.3 (Approximate performance difference for entropy-regularized NPG). *Suppose that we run NPG with step-size $0 < \eta_\theta \leq 1/\beta$, for any state $s_0 \in S$ we have that:*

$$V_r^{(t)}(s_0) - V_r^{(t+1)}(s_0) \leq 2\|\hat{\boldsymbol{Q}}_r^{(t)} - \boldsymbol{Q}_r^{(t)}\|_\infty.$$

This is a result derived in [Cen et al. 2021]. We refer the reader to appendix C.4 of the aforementioned work for proof.

Corollary 5.2.3.1 (Near monotone improvement of approximate NPG). *Under the Assumption $\|\hat{\boldsymbol{Q}}_r^{(t)} - \boldsymbol{Q}_r^{(t)}\|_\infty \leq \breve{\delta}$ we have that, for any $(s,a) \in S \times A$:*

$$Q_r^{(t)}(s,a) - Q_r^{(t+1)}(s,a) = \gamma \mathbb{E}_{s'|s,a}\big[V_r^{(t)}(s') - V_r^{(t+1)}(s')\big] \tag{5.28}$$

$$\leq 2\gamma\breve{\delta} \tag{5.29}$$

Lemma 5.2.4 (Stochastic NPG converges linearly to a neighborhood of the local optimum)**.** *We consider two sequences $\{\tilde{\boldsymbol{r}}^{(t)}\}_{t=0}^{T}$ and $\{\pi^{(t)}\}_{t=0}^{T}$ of diminished rewards and policies generated by Algorithm 3, with tabular softmax policy parameterization (Assumption (4.2.1)) and a linear reward class (Assumption (4.2.2)) using bounded mean euclidean distance error stochastic oracles (Assumptions 5.2.1 and 5.2.2) the policy converges to a local optimum (to a policy optimal w.r.t. $\tilde{\boldsymbol{r}}^{(t)}$) at rate:*

$$\mathbb{E}\Big[\|\boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}\|_{\infty}\Big] \leq C_1\lambda_1^t + \eta_z C_{\eta_z} + \breve{\delta}C_{\breve{\delta}}, \tag{5.30}$$

where:

$$\lambda_1 = 1 - (1-\gamma)\eta_\theta\beta, \tag{5.31}$$

$$C_1 = \frac{\gamma(4C_\lambda + 2\beta\log(mC_\lambda))}{(1-\eta_\theta\beta)(1+\gamma)-\gamma}, \tag{5.32}$$

$$C_\lambda = \|\Phi\|_1 + \|\Psi\|\lambda_{max} \tag{5.33}$$

$$C_{\eta_z} = \frac{2C_z^2\gamma}{\eta_\theta\beta(1-\gamma)}, \tag{5.34}$$

$$C_{\breve{\delta}} = \frac{\gamma(1-\eta_\theta 2(\gamma+1))}{\eta_\theta\beta(1-\gamma)}. \tag{5.35}$$

Proof. **Set up for the local convergence analysis**
Our analysis for the local convergence of *NPG-CIRL* is similar to the global convergence analysis of entropy-regularized *NPG* with perturbed gradients developed by [Cen et al. 2021]. It will rely on showing that the maximum Q-value error converges linearly to a small value, not directly by γ-contraction of an operator (as we did for Lemma 4.3.3) but through an affine system made up of three interdependent error quantities that all converge to small values.

Let us first define a few quantities relevant to our analysis, we let

$$\alpha = 1 - \eta_\theta\beta \tag{5.36}$$

and define the auxiliary sequence $\{\boldsymbol{\xi}^{(t)}\}_{t=0}^{T-1}$, $\boldsymbol{\xi}^{(t)} \in \mathbb{R}^{nm}$ recursively as follows:

$$\xi^{(0)}(s,a) := \|\boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(0)}}(s,\cdot)/\beta\|_1 \cdot \pi^0(a|s) \tag{5.37}$$

$$\xi^{(t+1)}(s,a) := \big(\xi^{(t)}(s,a)\big)^{\alpha} \exp\left(\frac{1-\alpha}{\beta}\hat{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a)\right). \tag{5.38}$$

Note that by definition this auxiliary sequence satisfies

$$\boldsymbol{\pi}^{(t)}(a|s) = \frac{\xi^{(t)}(s,a)}{\sum_{a'\in A}\xi^{(t)}(s,a')}, \tag{5.39}$$

for any iteration (t). The motivation for the definition of the sequence $\{\boldsymbol{\xi}^{(t)}\}_{t=0}^{T-1}$ might not be obvious, but it should become clear in the light of proposition 5.2.4 (contraction of the maximum Q-value error).

Now that we have defined the relevant quantities for our derivation, let us explicitly write our the quantities we will build our affine error system out of. We will consider three scalars $x_1^{(t)}, x_2^{(t)}, x_3^{(t)}$ that are defined for each iteration $t \in \{0,\dots,T\}$ of Algorithm 3:

$$x_1^{(t)} := \|\boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}\|_{\infty}, \tag{5.40}$$

$$x_2^{(t)} := \|\boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(t)}} - \beta\log\boldsymbol{\xi}^{(t)}\|_{\infty}, \tag{5.41}$$

$$x_3^{(t)} := -\min_{s,a}(\boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a) - \beta\log\boldsymbol{\xi}^{(t)}(s,a)). \tag{5.42}$$

The key idea here is that we will compute a "contraction-bound" for each term at iteration $t+1$ which will be a linear combination of all three terms at iteration t. For clarity, we organize those contraction rates intro three separate propositions (Propositions 5.2.4, 5.2.5 and 5.2.6) which we will then separately prove. Once this is done, we will move on to studying the affine system that naturally emerges from the propositions.

Proposition 5.2.4 (Contraction bound of the $x_1^{(t)}$ term)**.** *Under Assumptions 4.2.1, 4.2.2, 5.2.1 and 5.2.2, iterations of Algorithm 3 lead to the following contraction bound for error term $x_1^{(t+1)}$:*

$$\gamma\mathbb{E}_{(t+1)}[x_1^{(t+1)}] \leq \gamma(1-\alpha)x_1^{(t)} + \gamma\alpha x_2^{(t)} + \gamma\alpha x_3^{(t)} + \gamma(\breve{\delta}(2+2\gamma-\alpha) + 2\eta_z C_z^2). \tag{5.43}$$

The proof of this result is deferred to Appendix C.3.3.

Proposition 5.2.5 (Contraction bound of the $x_2^{(t)}$ term)**.** *Under Assumptions 4.2.1, 4.2.2, 5.2.1 and 5.2.2, iterations of Algorithm 2 lead to the following contraction bound for error term $x_2^{(t+1)}$:*

$$\mathbb{E}_{(t+1)}\left[x_2^{(t+1)}\right] \leq (1-\alpha)x_1^{(t)} + \alpha x_2^{(t)} + (1-\alpha)\breve{\delta} + \eta_z(C_z' + C_z\delta_z). \tag{5.44}$$

The proof of this result is deferred to Appendix C.3.1.

Proposition 5.2.6 (Contraction bound of the $x_3^{(t)}$ term)**.** *Under Assumptions 4.2.1, 4.2.2, 5.2.1 and 5.2.2, iterations of Algorithm 2 lead to the following contraction bound for error term $x_3^{(t+1)}$:*

$$\mathbb{E}_{(t+1)}[x_3^{(t+1)}] \leq \alpha x_3^{(t)} + (\breve{\delta}(1+2\gamma) + \eta_z(C_z' + C_z\delta_z). \tag{5.45}$$

The proof of this result is deferred to Appendix C.3.2.

Propositions 5.2.4, 5.2.5 and 5.2.6 naturally are grouped together as:

$$\overbrace{\mathbb{E}\left[\begin{bmatrix} x_1^{(t+1)} \\ x_2^{(t+1)} \\ x_3^{(t+1)} \end{bmatrix}\right]}^{:=\mathbb{E}[\boldsymbol{x}^{(t+1)}]} \leq \overbrace{\begin{bmatrix} \gamma(1-\alpha) & \gamma\alpha & \gamma\alpha \\ (1-\alpha) & \alpha & 0 \\ 0 & 0 & \alpha \end{bmatrix}}^{:=A} \overbrace{\begin{bmatrix} x_1^{(t)} \\ x_2^{(t)} \\ x_3^{(t)} \end{bmatrix}}^{:=\boldsymbol{x}^{(t)}} + \overbrace{\begin{bmatrix} \gamma(\breve{\delta}(2+2\gamma-\alpha) + 2\eta_z(C_z' + C_z\delta_z)) \\ (1-\alpha)\breve{\delta} + \eta_z(C_z' + C_z\delta_z) \\ \breve{\delta}(1+2\gamma) + \eta_z(C_z' + C_z\delta_z) \end{bmatrix}}^{:=b}. \tag{5.46}$$

The eigenvalues of the matrix A are the following:

$$\lambda_1 = \alpha + \gamma(1-\alpha) = 1 - (1-\gamma)\eta_\theta\beta, \qquad \lambda_2 = \alpha = 1 - \eta_\theta\beta, \qquad \lambda_3 = 0, \tag{5.47}$$

and they are associated with the following eigenvectors:

$$\boldsymbol{v}_1 = \begin{bmatrix} \gamma \\ 1 \\ 0 \end{bmatrix}, \qquad \boldsymbol{v}_2 = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}, \qquad \boldsymbol{v}_3 = \begin{bmatrix} \frac{\alpha}{1-\alpha} \\ 1 \\ 0 \end{bmatrix}. \tag{5.48}$$

Assume that our system starts at some point $\boldsymbol{x}^{(0)}$, and consider its representation into the basis formed by the eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3$:

$$\boldsymbol{x}^{(0)} = \sum_{i\in[3]} \boldsymbol{v_i} a_i. \tag{5.49}$$

The iterations of the affine system (5.46) can thus be bounded by:

$$\mathbb{E}[\boldsymbol{x}^{(t)}] = A\mathbb{E}[\boldsymbol{x}^{(t-1)}] + \boldsymbol{b} = (A)^t\boldsymbol{x}^{(0)} + \sum_{t=0}^{t-1}(A^t)\boldsymbol{b} \tag{5.50}$$

$$= \sum_{i\in[3]} \lambda_i^t \boldsymbol{v_i} a_i + \sum_{t=0}^{t-1}(A^t)\boldsymbol{b} \tag{5.51}$$

$$\overset{(i)}{\leq} \sum_{i\in[3]} \lambda_i^t \boldsymbol{v_i} a_i + \sum_{t=0}^{+\infty} (A^t)\boldsymbol{b} \tag{5.52}$$

$$\overset{(ii)}{=} \sum_{i\in[3]} \lambda_i^t \boldsymbol{v_i} a_i + (I-A)^{-1}\boldsymbol{b} \tag{5.53}$$

$$\overset{(iii)}{=} \lambda_1^t \begin{bmatrix}\gamma\\1\\0\end{bmatrix} a_1 + \lambda_2^t \begin{bmatrix}0\\-1\\1\end{bmatrix} a_2 + (I-A)^{-1}\boldsymbol{b} \tag{5.54}$$

where the inequality in (i) is element-wise true all eigenvalues of A are positive, equality (ii) is just a matrix geometric series. Note that in (iii), we omit the third eigenvector as its eigenvalues is 0. Explicit computation tells us that the coefficients a_1, a_2, a_3 are given by:

$$a_1 = \frac{1}{\alpha+\alpha\gamma-\gamma}\Big((\alpha-1)x_1^{(0)} + x_2^{(0)} + x_3^{(0)}\Big), \tag{5.55}$$

$$a_2 = x_3^{(0)}, \tag{5.56}$$

$$a_3 = \frac{1-\alpha}{\alpha+\alpha\gamma-\gamma}(x_1^{(0)} - \gamma x_2^{(0)} - \gamma x_3^{(0)}). \tag{5.57}$$

We mostly care about terms $x_1^{(t)}$ and $x_2^{(t)}$, direct computation yields, for the term $x_1^{(t)}$:

$$\mathbb{E}[x_1^{(t)}] \leq \frac{\gamma}{\alpha+\alpha\gamma-\gamma}\Big((\alpha-1)x_1^{(0)} + x_2^{(0)} + x_3^{(0)}\Big)\lambda_1^t + \eta_z \frac{2(C_z' + C_z\delta_z)\gamma}{(1-\alpha)(1-\gamma)} + \breve{\delta}\frac{\gamma(\alpha+2\gamma+2)}{(1-\alpha)(1-\gamma)} \tag{5.58}$$

$$= \frac{\gamma(-\eta_\theta\beta x_1^{(0)} + x_2^{(0)} + x_3^{(0)})}{\alpha+\alpha\gamma-\gamma}(1-(1-\gamma)\eta_\theta\beta)^t + \eta_z\frac{2(C_z' + C_z\delta_z)\gamma}{\eta_\theta\beta(1-\gamma)} + \breve{\delta}\frac{\gamma(1-\eta_\theta 2(\gamma+1))}{\eta_\theta\beta(1-\gamma)}, \tag{5.59}$$

$$\leq \frac{\gamma(4C_\lambda + 2\beta\log(mC_\lambda))}{(1-\eta_\theta\beta)(1+\gamma)-\gamma}(1-(1-\gamma)\eta_\theta\beta)^t + \eta_z\frac{2(C_z' + C_z\delta_z)\gamma}{\eta_\theta\beta(1-\gamma)} + \breve{\delta}\frac{\gamma(1-\eta_\theta 2(\gamma+1))}{\eta_\theta\beta(1-\gamma)}, \tag{5.60}$$

where $C_\lambda = \|\Phi\|_1 + \|\Psi\|\lambda_{\max}$. This thus gives a convergence rate for Q-value suboptimality as $x_1^{(t)} = \|\boldsymbol{Q}^*_{\tilde{r}^{(t+1)}} - \boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t+1)}}\|_\infty$:

$$\|\boldsymbol{Q}^*_{\tilde{r}^{(t+1)}} - \boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t+1)}}\|_\infty \leq C_1\lambda_1^t + \eta_z C_{\eta_z,1} + \breve{\delta}C_{\breve{\delta},1}. \tag{5.61}$$

On the other hand, the term $x_2^{(t)}$ converges at rate:

$$\mathbb{E}[x_2^{(t)}] \leq \frac{(\alpha-1)x_1^{(0)} + x_2^{(0)} + x_3^{(0)}}{\alpha+\alpha\gamma-\gamma}\big(1-(1-\gamma)\eta_\theta\beta\big)^t - x_3^{(0)}\big(1-\eta_\theta\beta\big)^t \tag{5.62}$$
$$+ \eta_z\frac{(C_z' + C_z\delta_z)(1+\gamma)}{(1-\alpha)(1-\gamma)} + \breve{\delta}\frac{2\gamma^2+1+\gamma-\alpha}{(1-\alpha)(1-\gamma)}$$
$$\leq \frac{\gamma(4C_\lambda + 2\beta\log(mC_\lambda))}{(1-\eta_\theta\beta)(1+\gamma)-\gamma}\big(1-(1-\gamma)\eta_\theta\beta\big)^t - x_3^{(0)}\big(1-\eta_\theta\beta\big)^t \tag{5.63}$$
$$+ \eta_z\frac{(C_z' + C_z\delta_z)(1+\gamma)}{\eta_\theta\beta(1-\gamma)} + \breve{\delta}\frac{2\gamma^2+\gamma+\eta_\theta\beta}{\eta_\theta\beta(1-\gamma)}$$

The convergence rate of the term $x_2^{(t)}$ provides a convergence rate for the log-difference of the policies since:

$$\|\log\boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}^{(t)}} - \log\boldsymbol{\pi}^{(t)}\|_\infty \leq 2\|\log\boldsymbol{Q}_{\tilde{\boldsymbol{r}}^{(t)}}/\beta - \log\boldsymbol{\xi}^{(t)}\|_\infty = \frac{2}{\beta}x_2^{(t)}, \tag{5.64}$$

we thus have:

$$\|\log\boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}^{(t)}} - \log\boldsymbol{\pi}^{(t)}\|_\infty \leq C_1\lambda_1^t + C_2\lambda_2^t + \eta_z C_{\eta_z,2} + \breve{\delta}C_{\breve{\delta},2}. \tag{5.65}$$

We are done with the proof of Lemma 5.2.4 □

5.2.5. Global convergence

With local convergence established, we turn our attention to showing global convergence of Algorithm 3 (Theorem 5.2.1). To do so we follow a similar approach to the one used to prove Theorem 4.3.1. Concretely, we use that the local convergence provides an approximation of the dual D to problem (P2).

Proof. We start by decomposing the suboptimality of the function L:

$$L(\boldsymbol{\theta}, \boldsymbol{z}) - L^* = \overbrace{D(\boldsymbol{z}) - L^*}^{(a)} + \overbrace{L(\boldsymbol{\theta}, \boldsymbol{z}) - D(\boldsymbol{z})}^{(b)}. \tag{5.66}$$

We will first consider term (b) which implies that L approximates the dual and then we consider term (a) which implies that gradient descent on the approximate dual converges. Using the soft suboptimally (Lemma 2.4.3) we have:

$$\left|L(\boldsymbol{\theta}^{(T)}, \boldsymbol{z}^{(T)}) - D(\boldsymbol{z}^{(T)})\right| \leq \frac{2}{\eta_\theta} \| \log \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}} - \log \boldsymbol{\pi}^{(t+1)}_{\tilde{\boldsymbol{r}}} \|_\infty. \tag{5.67}$$

We already have shown this result when proving global with exact gradients[4], so we omit the details of the bound. Equation (5.67) provides a way to show that Lemma 5.2.4 implies that we reach an approximation of the dual:

$$\mathbb{E}\Big[\left|L(\boldsymbol{\theta}^{(T)}, \boldsymbol{z}^{(T)}) - D(\boldsymbol{z}^{(T)})\right|\Big] \leq \frac{2}{\eta_\theta}\Big(C_1 \lambda_1^t + C_2 \lambda_2^t + \eta_z C_{\eta_z, 2} + \breve{\delta} C_{\breve{\delta}, 2}\Big). \tag{5.68}$$

We thus move on to studying the dual suboptimally term (a), where we will follow the same general steps as in the exact gradient setting: First we separate our gradient $\boldsymbol{g}_z$ into three terms:

$$\boldsymbol{g}_z^{(t)} = \nabla_z L(\boldsymbol{\theta}^{(t)}, \boldsymbol{z}^{(t)}) + \boldsymbol{b}_z^{(t)} = \nabla_z D(\boldsymbol{\theta}^{(t)}, \boldsymbol{z}^{(t)}) + \boldsymbol{\sigma}_z^{(t)} + \boldsymbol{b}_z^{(t)}, \tag{5.69}$$

where $\boldsymbol{\sigma}_z^{(t)} = \nabla_z D(\boldsymbol{\theta}^{(t)}, \boldsymbol{z}^{(t)}) - \nabla_z L(\boldsymbol{\theta}^{(t)}, \boldsymbol{z}^{(t)})$ is the dual-approximation error and $\boldsymbol{b}_z^{(t)} = \nabla_z L(\boldsymbol{\theta}^{(t)}, \boldsymbol{z}^{(t)}) - \boldsymbol{g}_z^{(t)}$ is the error of the gradient orcales. Using Assumption 5.2.2, we know that $\|\boldsymbol{b}_z^{(t)}\|_2 \leq \delta_z$. Rearranging (5.69) and using convexity of the dual function D we get:

$$\begin{aligned} D(\boldsymbol{z}^{(t)}) - D^* &\leq \langle \nabla_z D(\boldsymbol{z}^{(t)}), \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle = \mathbb{E}\big[\langle \boldsymbol{g}_z^{(t)} - \boldsymbol{\sigma}_z^{(t)} - \boldsymbol{b}_z^{(t)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle\big] && (5.70) \\ &\leq \overbrace{\mathbb{E}\big[\langle \boldsymbol{g}_z^{(t)} \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle\big]}^{(A)} + \overbrace{\Big|\mathbb{E}\big[\langle \boldsymbol{\sigma}_z^{(t)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle\big]\Big|}^{(B)} + \overbrace{\Big|\mathbb{E}\big[\langle \boldsymbol{b}_z^{(t)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle\big]\Big|}^{(C)}. && (5.71) \end{aligned}$$

We thus have three terms: (A) the algorithm step term, (B) the dual-approximation error term and (C) the orcale error term. We will show that (B) and (C) are sufficiently small and then study the convergence of the perturbed algorithm by looking at (A). We start by considering term (B):

$$\begin{aligned} \Big|\mathbb{E}\big[\langle \boldsymbol{\sigma}_z^{(t)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle\big]\Big| &\overset{(i)}{\leq} D_{1,z} \mathbb{E}[\|\boldsymbol{\sigma}_z^{(t)}\|_\infty] && (5.72) \\ &\overset{(ii)}{\leq} \frac{C_z D_{1,z} \sqrt{f+d}(1+\gamma\sqrt{nm})}{2} \big\| \log \boldsymbol{\pi}^{(t)} - \log \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}} \big\|_\infty && (5.73) \\ &\overset{(iii)}{\leq} \frac{C_z D_{1,z} \sqrt{f+d}(1+\gamma\sqrt{nm})}{2} \Big(C_1 \lambda_1^t + C_2 \lambda_2^t + \eta_z C_{\eta_z, 2} + \breve{\delta} C_{\breve{\delta}, 2}\Big), && (5.74) \end{aligned}$$

where (i) is obtained by applying Hölder's inequality and plugging in the diameter of the z variables in the $\|\cdot\|_1$ norm $C_{1,z}$. Furthermore, we obtain (ii) through the same exact derivation as (4.78) and (iii) from plugging the policy convergence bound of Lemma 5.2.4. Term (C) of (5.71) can be bounded as follows:

$$\Big|\mathbb{E}\big[\langle \boldsymbol{b}_z^{(t)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle\big]\Big| \overset{(i)}{\leq} D_{1,z} \mathbb{E}[\|\boldsymbol{b}_z^{(t)}\|_\infty] \overset{(ii)}{\leq} D_{1,z} \mathbb{E}[\|\boldsymbol{b}_z^{(t)}\|_2] \overset{(iii)}{\leq} D_{1,z} \delta_z, \tag{5.75}$$

[4] See equation (4.54).

where (i) is obtained by applying Hölder's inequality and by plugging in $C_{1,z}$, the diameter of the z variables in the $\|\cdot\|_1$ norm). Using in (ii) that the $\|\cdot\|_2$ norm upper-bounds the $\|\cdot\|_\infty$ norm and in (iii) that Assumption 5.2.2 holds gives us our result. We are now ready to consider term (A) and study global convergence:

$$\mathbb{E}\big[\langle \boldsymbol{g}_z^{(t)} \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle\big] \leq \frac{1}{2\eta_z}\Big(\mathbb{E}\big[\eta_z^2 \|\boldsymbol{g}^{(t)}\|_2^2\big] + \mathbb{E}\big[\|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2^2\big] - \mathbb{E}\big[\|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^*\|_2^2\big]\Big). \tag{5.76}$$

Inserting (5.74), (5.75) and (5.76) into (5.71) we get:

$$\begin{aligned}
\mathbb{E}\big[D(\boldsymbol{z}^{(t)}) - D^*\big] \leq & \frac{1}{2\eta_z}\Big(\mathbb{E}\big[\eta_z^2 \|\boldsymbol{g}^{(t)}\|_2^2\big] + \mathbb{E}\big[\|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2^2\big] - \mathbb{E}\big[\|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^*\|_2^2\big]\Big) && (5.77)\\
& + \frac{C_z D_{1,z}\sqrt{f+d}(1+\gamma\sqrt{nm})}{2}\Big(C_1\lambda_1^t + C_2\lambda_2^t + \eta_z C_{\eta_z,2} + \breve{\delta} C_{\breve{\delta},2}\Big) + D_{1,z}\delta_z \\
= & \frac{1}{2\eta_z}\Big(\mathbb{E}\big[\eta_z^2 \|\boldsymbol{g}^{(t)}\|_2^2\big] + \mathbb{E}\big[\|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2^2\big] - \mathbb{E}\big[\|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^*\|_2^2\big]\Big) && (5.78)\\
& + C_l C_1 \lambda_1^t + C_l C_2 \lambda_2^t + C_l C_{\eta_z,2}\eta_z + C_L C_{\breve{\delta},2}\breve{\delta} + D_{1,z}\delta_z.
\end{aligned}$$

Taking an empirical average across T algorithm steps we get:

$$\begin{aligned}
\mathbb{E}\frac{1}{T}\left[\sum_{t=1}^{T} D(\boldsymbol{z}^{(t)})\right] - D^* \leq & \frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{2\eta_z}\Big(\mathbb{E}\big[\eta_z^2 \|\boldsymbol{g}^{(t)}\|_2^2\big] + \mathbb{E}\big[\|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2^2\big] - \mathbb{E}\big[\|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^*\|_2^2\big]\Big) && (5.79)\\
& + \frac{1}{T}\sum_{t=0}^{T-1}\Big(C_l C_1 \lambda_1^t + C_l C_2 \lambda_2^t + C_l C_{\eta_z,2}\eta_z + C_L C_{\breve{\delta},2}\breve{\delta} D_{1,z}\delta_z\Big) \\
= & \frac{\eta_z}{2T}\sum_{t=0}^{T-1}\mathbb{E}\big[\|\boldsymbol{g}^{(t)}\|_2^2\big] + \frac{1}{2\eta_z T}\Big(+\mathbb{E}\big[\|\boldsymbol{z}^{(0)} - \boldsymbol{z}^*\|_2^2\big] - \mathbb{E}\big[\|\boldsymbol{z}^{(T)} - \boldsymbol{z}^*\|_2^2\big]\Big) && (5.80)\\
& + \frac{C_l}{T}\left(C_1\sum_{t=0}^{T-1}\lambda_1^t + C_2\sum_{t=0}^{T-1}\lambda_2^t\right) + \frac{1}{T}\sum_{t=0}^{T-1}(C_l C_{\eta_z,2}\eta_z + C_L C_{\breve{\delta},2}\breve{\delta} + D_{1,z}\delta_z) \\
\leq & \frac{\eta_z}{2T}\sum_{t=0}^{T-1}\mathbb{E}\big[\|\boldsymbol{g}^{(t)}\|_2^2\big] + \frac{\|\boldsymbol{z}^{(0)} - \boldsymbol{z}^*\|_2^2}{2\eta_z T} && (5.81)\\
& + \frac{C_l}{T}\left(\frac{C_1}{1-\lambda_1} + \frac{C_2}{1-\lambda_2}\right) + \big(C_l C_{\eta_z,2}\eta_z + C_L C_{\breve{\delta},2}\breve{\delta} + D_{1,z}\delta_z\big) \\
\leq & \frac{\eta_z(\|\Psi\|^2 + \|\Phi\|^2 + \|\boldsymbol{b}\|_2^2 + \delta_z^2)}{2} + \frac{\|\boldsymbol{z}^{(0)} - \boldsymbol{z}^*\|_2^2}{2\eta_z T} && (5.82)\\
& + \frac{C_l}{T}\left(\frac{C_1}{1-\lambda_1} + \frac{C_2}{1-\lambda_2}\right) + \big(C_l C_{\eta_z,2}\eta_z + C_L C_{\breve{\delta},2}\breve{\delta} + D_{1,z}\delta_z\big) \\
\leq & \frac{\|\Psi\|^2 + \|\Phi\|^2 + \|\boldsymbol{b}\|_2^2 + \delta_z^2}{2T^u} + \frac{\|\boldsymbol{z}^{(0)} - \boldsymbol{z}^*\|_2^2}{2T^{1-u}} && (5.83)\\
& + \frac{C_l}{T}\left(\frac{C_1}{1-\lambda_1} + \frac{C_2}{1-\lambda_2}\right) + \Big(\frac{C_l C_{\eta_z,2}}{T^u} + C_L C_{\breve{\delta},2}\breve{\delta} + D_{1,z}\delta_z\Big)
\end{aligned}$$

Choosing the optimal $u = 1/2$ as in (4.89) and bringing back the upper bound on the dual approximation term $|L(\boldsymbol{\theta}^{(t)}, \boldsymbol{z}^{(t)}) - D(\boldsymbol{z}^{(t)})|$ from (5.68) we reach the expression of our optimal convergence rate:

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} L(\boldsymbol{z}^{(t)}, \boldsymbol{\theta}^{(t)})\right] - L^* \leq & \left(\frac{\|\Psi\|^2 + \|\Phi\|^2 + \|\boldsymbol{b}\|_2^2 + \delta_z^2 + \|\boldsymbol{z}^{(0)} - \boldsymbol{z}^*\|_2^2}{2} + C_l C_{\eta_z,2} + C_{\eta_z,2}\right)\frac{1}{\sqrt{T}} && (5.84)\\
& + C_l\left(\frac{C_1}{1-\lambda_1} + \frac{C_2}{1-\lambda_2}\right)\frac{1}{T} + \frac{2}{\eta_\theta}\Big(C_1\lambda_1^t + C_2\lambda_2^t\Big)
\end{aligned}$$

$$
\begin{aligned}
&+ (C_L + 1)C_{\delta,2}\delta + D_{1,z}\delta_z \\
&= C_{\text{sg},1}\frac{1}{\sqrt{T}} + C_{\text{sg},2}\frac{1}{T} + \frac{2}{\eta_\theta}\Big(C_1\lambda_1^t + C_2\lambda_2^t\Big) + (C_L + 1)C_{\delta,2}\delta + D_{1,z}\delta_z.
\end{aligned} \tag{5.85}
$$

□

5.2.6. Sample complexity

The sample complexity result is a corollary of Theorem 5.2.1. Recall that the theorem states that the global convergence rate of Algorithm 3 is:

$$
\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} L(\boldsymbol{z}^{(t)}, \boldsymbol{\theta}^{(t)})\right] - L^* = O\Big(\frac{1}{\sqrt{T}} + \delta + \delta_z\Big). \tag{5.86}
$$

Reaching an ϵ-approximation of the optimal solution thus requires ensuring that $\delta < \epsilon$ and that $\delta_z < \epsilon$. We will thus need to upper bound the quantities δ and δ_z explicitly. We have already done so with Propositions 5.2.1, 5.2.2 and 5.2.3. Specifically, when using the GPOMDP estimator (assumptions 5.1.1), by Proposition 5.2.1, the mean euclidean distance error decreases at rate:

$$
\delta = \mathbb{E}\Big[\|\boldsymbol{g}_\theta^{(t)} - \nabla_\theta L(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\lambda})\|_2\Big] = O\Big(H\gamma^H + \frac{1}{B}\Big). \tag{5.87}
$$

This implies that we can ensure Assumption 5.2.1 (bounded mean Euclidean distance error for the policy gradient oracle) is satisfied by sampling batches of size and horizon:

$$
H = O(\log(\epsilon^{-1})) \qquad\qquad B = O(\epsilon^{-1}). \tag{5.88}
$$

Similarly, Proposition 5.2.2 implies that the bounded mean Euclidean distance error of the reward gradient estimator decreases at rate:

$$
\mathbb{E}\Big[\|\boldsymbol{g}_r^{(t)} - \nabla_r L(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\lambda})\|_2\Big] = O\Big(\gamma^H + \frac{1}{B}\Big), \tag{5.89}
$$

the same result can be stated about the Lagrangian multiplier gradient estimator by proposition 5.2.3:

$$
\mathbb{E}\Big[\|\boldsymbol{g}_\lambda^{(t)} - \nabla_\lambda L(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\lambda})\|_2\Big] = O\Big(\gamma^H + \frac{1}{B}\Big). \tag{5.90}
$$

Bringing together (5.89) and (5.2.3) allows us to get a sample complexity result for the z gradient estimator. We can therefore ensure that Assumption 5.2.2 is satisfied with sample complexity:

$$
H = O(\log(\epsilon^{-1})) \qquad\qquad B = O(\epsilon^{-1}). \tag{5.91}
$$

This leads us to the same sample complexity result as the one concerning the policy gradient estimator. For convergence to an ϵ error, we need to run $O(\epsilon^{-2})$ iterations of the algorithm with $O(\epsilon^{-2})$ samples per iteration. The complete algorithm thus has the following sample complexity:

$$
T = O\left(\frac{1}{\epsilon^4}\right). \tag{5.92}
$$

One aspect that we do not consider in our analysis and that might be worth studying is the cost the computational of estimating the Fisher information matrix.

6

Conditions for convergence of NPG-CIRL

In the following Chapter, we discuss structural assumptions on the NPG problem that might allow for faster convergence rates of the algorithm. We then describe a condition that, when satisfied, ensures that the problem converges linearly fast.

6.1. Structure of the NPG-CIRL problem, dual strong convexity

The core idea of our analysis is to leverage linear convergence of gradient descent over strongly convex, smooth functions to show linear convergence of the gradient descent-ascent scheme of Algorithm 2. We observe that when the right conditions are met (strong convexity and smoothness of the dual problem), we can construct a single affine error system to show global convergence. This analysis, similar to the one establishing the local convergence lemma of Chapter 5 (Lemma 4.3.3) immediately shows that the convergence rate is linear.

6.1.1. Assumptions

We rely on the same assumptions as the results from Sections 4 and 5: we assume tabular softmax parametrization (Assumption 4.2.1), a linear reward class (Assumption 4.2.2), negative Shannon entropy regularization (Assumption 4.2.3) and a non-vanishing occupancy measure (Assumption 4.2.4). Furthermore, we introduce two assumptions on the dual function D required for fast convergence: dual smoothness and dual strong convexity.

Assumption 6.1.1 (Strong convexity of the dual)**.** *The dual function of the Lagrangian L defined in (3.15) is strongly-convex with constant C_{SC}.*

Assumption 6.1.2 (Dual smoothness)**.** *The dual function D of the Lagrangian (P2) is differentiable and smooth with parameter L_z.*

These two assumptions may seem arbitrary, but we argue they are reasonable on MDPs that meet specific properties. We discuss this in detail in Appendix D.1.

6.2. Fast-convergence dynamics

We now discuss the gradient descent dynamics at play. When Assumptions 6.1.1 and 6.1.2 are met. It is possible to build an affine error system that directly describes the global convergence of the algorithm. Consider the three following scalars that quantify error:

$$x_1^{(t)} = \|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2 \tag{6.1}$$

$$x_2^{(t)} = \|\boldsymbol{Q}^{(t)}_{\boldsymbol{r}^{\tilde{(t)}}} - \boldsymbol{Q}^{*}_{\boldsymbol{r}^{\tilde{(t)}}}\|_\infty \tag{6.2}$$

$$x_3^{(t)} = \|\boldsymbol{Q}^{*}_{\boldsymbol{r}^{\tilde{(t)}}} - \beta \log \boldsymbol{\xi}^{(t)}\|_\infty \tag{6.3}$$

Let us define a few relevant quantities relevant to our analysis. First, we let the constant C be given by:

$$C = \max\left\{\frac{\gamma C_z L_z}{2b}, \frac{C_z L_z}{4b}, C_\sigma\right\}. \tag{6.4}$$

Where $C_z = 2(\|\Psi\| + \|\Psi\|)$, L_z is the smoothness constant from Assumption D.1.2, is some real number $b > 0$ and C_σ is a constant related to the properties of the *MDP* (see Appendix D.2.3). We introduce the normalized learning rates:

$$\tilde{\eta}_z = \frac{\eta_z}{C_{\text{SC}}} \in (0, 1], \qquad \tilde{\eta}_\theta = \frac{\eta_\theta}{\beta} \in (0, 1]. \tag{6.5}$$

As well as the system matrix:

$$K = \begin{bmatrix} (1-\tilde{\eta}_z) & 0 & C\tilde{\eta}_z \\ C\tilde{\eta}_z & \gamma & 2\gamma(1-\tilde{\eta}_\theta) \\ C\tilde{\eta}_z & \tilde{\eta}_\theta & (1-\tilde{\eta}_\theta) \end{bmatrix}. \tag{6.6}$$

Lemma 6.2.1 (Affine error system for fast convergence)**.** *When Assumptions 4.2.1 (softmax policy), 4.2.2 (linear reward class), 6.1.1 (dual strong convexity) and 6.1.2 (dual smoothness) are satisfied, then, for any $b > 0$, iterations of Algorithm 2 satisfy:*

$$\boldsymbol{x}^{(t+1)} \leq K\boldsymbol{x}^{(t)} + \boldsymbol{b}. \tag{6.7}$$

Where K is the system matrix as defined in equation (6.6), $\boldsymbol{x}^{(t)} = [x_1^{(t)}, x_2^{(t)}, x_3^{(t)}]^\top$ is the error vector (which is defined for any iteration t of Algorithm 2) and $\boldsymbol{b} = [0, 2\gamma b, b]^\top$. The proof of this result is deferred to Appendix D.2

Lemma 6.2.1 naturally allows for the identification of a condition for fast convergence of our algorithm.

Condition 6.2.1. $\|K\| < 1$ *(the system matrix has a spectral norm strictly inferior to 1).*

Theorem 6.2.2 (Linear convergence when condition 6.2.1 is met)**.** *When Assumptions 4.2.1 (softmax policy), 4.2.2 (linear reward class), 6.1.1 (dual strong convexity) and 6.1.2 (dual smoothness) are satisfied, and when Condition 6.2.1 is met, then Algorithm 2 to reaches an error:*

$$\left|L(\boldsymbol{\theta}^{(T)}, \boldsymbol{w}^{(T)}, \boldsymbol{\lambda}^{(T)}) - L^*\right| \leq \epsilon, \tag{6.8}$$

in: $T = O(\log(1/\epsilon))$, iterations.

6.3. Discussion

Next, we discuss when Condition 6.2.1 is met.

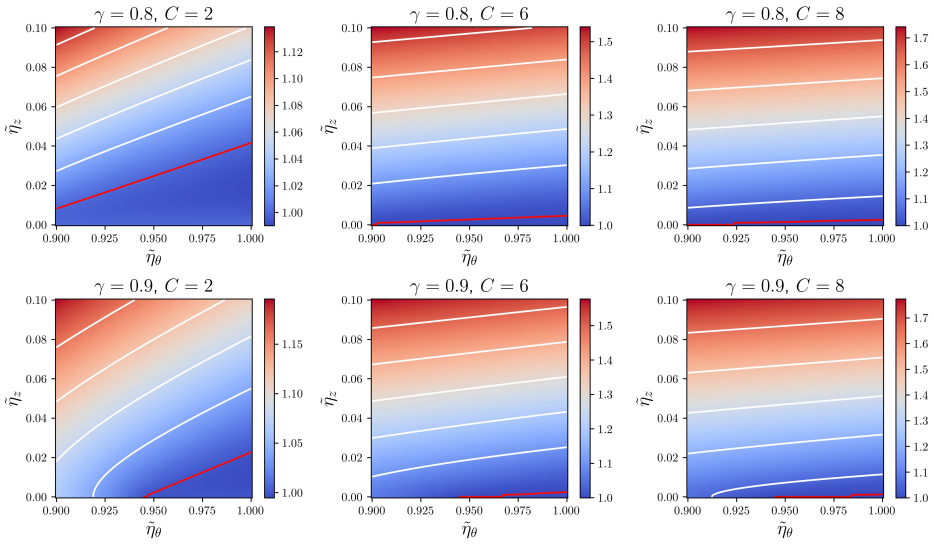


Figure 6.1: Spectral norms of the K matrix as a function of both learning rates ($\tilde{\eta}_\theta$ on the x-axis and $\tilde{\eta}_z$ on the y axis) for different values of γ and of C. The red-line denote the values of $\tilde{\eta}_\theta$ and $\tilde{\eta}_z$ for which $\|A\|$ is exactly 1. The plot shows, the importance of the constants γ and C. Note that the plots show the domain $\tilde{\eta}_z \in [0, 0.1]$, $\tilde{\eta}_\theta \in [0.9, 1]$.

Condition 6.2.1 provides a way to describe situations in which algorithm 2 converges linearly fast, but it is hard to establish whether this condition is a reasonable assumption to make about the structure of the optimization problem.

The spectral norm $\|K\|$ can be thought of as a function of the primal and dual normalized learning rates $\tilde{\eta}_\theta$ and $\tilde{\eta}_z$ as well as of the system dependant constant C and the discount factor γ. One way to think about the condition $\|K\|$ is as a way of specifying which combinations of learning rates $\tilde{\eta}_\theta$, $\tilde{\eta}_z$ lead to linear convergence assuming that C and γ are given. To provide some intuition and investigate whether linear convergence can practically happen when running iterations of Algorithm 2, we produced plots of the value of $\|K\|$ as functions of both learning rates $\tilde{\eta}_\theta$, $\tilde{\eta}_z$ (Figure 6.1).

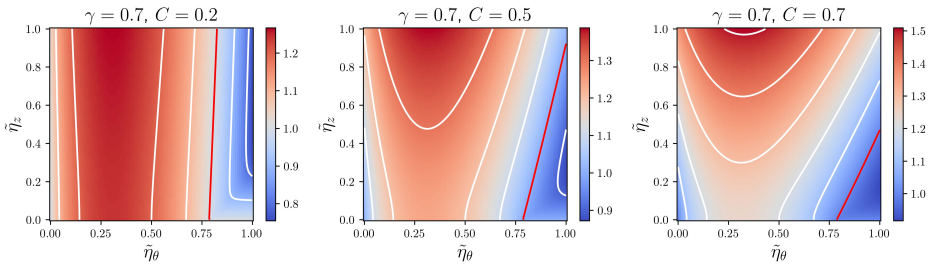


Figure 6.2: Spectral norms of the K matrix as a function of both learning rates ($\tilde{\eta}_\theta$ on the x-axis and $\tilde{\eta}_z$ on the y axis) low values of C. The red-line denote the values of $\tilde{\eta}_\theta$ and $\tilde{\eta}_z$ for which $\|A\|$ is exactly 1. The plot shows, the importance of the constants γ and C. Note that the plots show the domain $\tilde{\eta}_z \in [0, 1]$, $\tilde{\eta}_\theta \in [0, 1]$.

What the resulting plots highlight (Figure 6.1) is that when the constants C and γ are sufficiently small, then a region of the $\tilde{\eta}_\theta$, $\tilde{\eta}_z$ space indeed satisfies the condition. Specifically the condition is met when $\tilde{\eta}_\theta$ is close to 1 and when $\tilde{\eta}_z$ is close 0. This suggests that what happens is akin to dual descent, where the optimizer moves faster to a locally optimal policy with respect to $\boldsymbol{\theta}$ than it does to the global optimum. This behaviour is similar to what we analyzed in Chapters 4 and 5.

On the other hand, when the constant C is sufficiently smaller than 1, we observe that we might reach a situation where linear convergence happens with a learning rate $\tilde{\eta}_z$ close to 1 (Figure 6.2). This is the ideal setting for fast convergence. Devising a way to reach lower the constant might be critical to the design of a fast converging algorithm. Our derivation suggests that this can be reached through the choice of the right feature matrices, i.e. when the quantity $\|\Phi\| + \|\Psi\|$ is sufficiently small. Whether this is attainable in practice remains to be verified.

7

Conclusion

In conclusion, three main results are obtained in this work.

First, we propose a method for solving the *CIRL* problem, termed *NPG-CIRL*. We motivate the choices made in the design of the *NPG-CIRL* algorithm and discuss possible tradeoffs in implementation. We first discuss an idealized version of the algorithm, where Oracle access to exact gradients is available (we call that setting the *exact gradient setting*). We then move away from that idealized setting and show how our algorithm can be more practically implemented using Monte-Carlo gradient estimation techniques (we call that setting the *stochastic gradient setting*).

Secondly, we provide a finite-time analysis of the global convergence of our algorithm in the *exact gradient setting*. We show that the algorithm requires $O(1/\epsilon^2)$ gradient evaluations to reach an ϵ approximate solution (this is the *iteration complexity* of our algorithm). We also show that the recovered policy provably satisfies the imposed constraints and, thus, that our algorithm is safe.

Thirdly we show that the convergence rate obtained in the idealized *exact gradient setting* still holds when using biased, stochastic gradient estimators. The stochastic algorithm still requires $O(1/\epsilon^2)$ gradient evaluations to converge. We extend our analysis and quantify the total required number of samples (*MDP steps*) needed to reach convergence to an ϵ-approximate solution and show that our algorithm requires $O(1/\epsilon^4)$ samples (this is the *sample complexity* of our algorithm).

7.0.1. Future work

While this work succeeds in proposing and analysing a globally converging algorithm to tackle the *CIRL* problem, several important questions remain unanswered.

Firstly we believe running a set of experiments using the stochastic formulation of our algorithm in a practical application would be of the most significant interest. We only theoretically analysed our algorithm in the discrete, tabular softmax setting. However, it resembles other policy-gradient algorithms that perform well when using deep neural networks to parametrise the policies. Running a set of experiments using neural networks to estimate the policies and rewards would thus be of great interest.

Running a set of experiments would also help build a better qualitative understanding of the algorithm's behaviour. It would showcase the safety guarantees provided by the algorithm, highlight the explainability benefits of *CIRL*, and maybe emphasise certain aspects that were overlooked when studying it from a theoretical viewpoint.

Secondly, we believe that further theoretical investigation into the conditions for linear convergence of *NPG-CIRL* may yield significant results. The fast convergence dynamics described in Chapter 6 suggest that there may be a way of modifying *NPG-CIRL* to ensure that it converges fast reliably. If so, it would be of great interest to study this algorithm in the stochastic setting. The investigation

of linear convergence would also benefit from running a set of experiments to investigate if this behaviour can realistically be achieved in real-world problems.

References

Abbeel, Pieter and Andrew Y. Ng (2004). "Apprenticeship learning via inverse reinforcement learning". In: *ICML '04: Proceedings of the twenty-first international conference on Machine learning*. Banff, Alberta, Canada: ACM. ISBN: 1-58113-828-5.

Achiam, Joshua et al. (June 2017). "Constrained Policy Optimization". In: ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 22–31. URL: https://proceedings.mlr.press/v70/achiam17a.html.

Agarwal, Alekh et al. (2020). "On the Theory of Policy Gradient Methods: Optimality, Approximation, and Distribution Shift". In: *Proceedings of Machine Learning Research*.

Altman, Eitan (1999). "Constrained Markov Decision Processes". In.

Baxter, Jonathan and Peter L. Bartlett (2001). "Infinite-horizon policy-gradient estimation". In: *Journal of Artificial Intelligence Research*.

Bellman, Richard (1957). "A Markovian Decision Process". In: *Journal of Mathematics and Mechanics* 6.5, pp. 679–684. ISSN: 00959057, 19435274. URL: http://www.jstor.org/stable/24900506 (visited on 06/25/2023).

Boyd, Stephen and Lieven Vandenberghe (2004). *Convex optimization*. Cambridge university press.

Bubeck, Sébastien (2015). "Convex Optimization: Algorithms and Complexity". In: URL: http://arxiv.org/abs/1405.4980.

Cen, Shicong et al. (2021). "Fast Global Convergence of Natural Policy Gradient Methods with Entropy Regularization". In: *Operations Research*.

Ding, Dongsheng et al. (2020). "Natural Policy Gradient Primal-Dual Method for Constrained Markov Decision Processes". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33, pp. 8378–8390.

Ding, Fan and Yexiang Xue (2022). "X-MEN: guaranteed XOR-maximum entropy constrained inverse reinforcement learning". In: *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*. Ed. by James Cussens and Kun Zhang. Vol. 180. Proceedings of Machine Learning Research. PMLR.

Ding, Yuhao, Junzi Zhang, and Javad Lavaei (2021). "Beyond Exact Gradients: Convergence of Stochastic Soft-Max Policy Gradient Methods with Entropy Regularization". In: *ArXiv* abs/2110.10117.

Geist, Matthieu, Bruno Scherrer, and Olivier Pietquin (2019). "A Theory of Regularized Markov Decision Processes". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 2160–2169. URL: https://proceedings.mlr.press/v97/geist19a.html.

Guigues, Vincent (Apr. 2020a). "Inexact stochastic mirror descent for two-stage nonlinear stochastic programs". In: *Mathematical Programming* 187. DOI: 10.1007/s10107-020-01490-5.

— (June 2020b). *On the strong concavity of the dual function of an optimization problem*.

Haarnoja, Tuomas et al. (2018). "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor." In: *ICML*. Ed. by Jennifer G. Dy and Andreas Krause. Proceedings of Machine Learning Research. PMLR.

Kakade, Sham M. (2001). "A Natural Policy Gradient". In: *NIPS*.

Kiran, Ravi et al. (2022). "Deep Reinforcement Learning for Autonomous Driving: A Survey". In: *IEEE Transactions on Intelligent Transportation Systems* 23.6, pp. 4909–4926.

Lee, Joonho et al. (2020). "Learning quadrupedal locomotion over challenging terrain". In: *Science Robotics* 5.47, eabc5986.

Mankowitz, Daniel J. et al. (2023). "Faster sorting algorithms discovered using deep reinforcement learning". In: *Nature* 618.7964. DOI: 10.1038/s41586-023-06004-9.

Mei, Jincheng et al. (2020). "On the Global Convergence Rates of Softmax Policy Gradient Methods". In: *Proceedings of the 37th International Conference on Machine Learning*. ICML'20. JMLR.org.

Nachum, Ofir et al. (2017). "Bridging the Gap between Value and Policy Based Reinforcement Learning". In: *Advances in Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc.

Ng, A., Daishi Harada, and Stuart J. Russell (1999). "Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping". In: *International Conference on Machine Learning*.

Paternain, Santiago, Miguel Calvo-Fullana, et al. (2023). "Safe Policies for Reinforcement Learning via Primal-Dual Methods". In: *IEEE Transactions on Automatic Control* 68.3. DOI: 10.1109/TAC.2022.3152724.

Paternain, Santiago, Luiz Chamon, et al. (2019). "Constrained Reinforcement Learning Has Zero Duality Gap". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Curran Associates, Inc.

Peng, Xue Bin et al. (July 2020). "Learning Agile Robotic Locomotion Skills by Imitating Animals". In: *Robotics: Science and Systems*. DOI: 10.15607/RSS.2020.XVI.064.

Puterman, Martin L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons.

Ramachandran, Deepak and Eyal Amir (Jan. 2007). "Bayesian Inverse Reinforcement Learning." In: pp. 2586–2591.

Rockafellar, R. Tyrrell and Roger J.-B. Wets (1998). *Variational Analysis*. Heidelberg, Berlin, New York: Springer Verlag.

Russell, Stuart J. (1998). "Learning agents for uncertain environments". In: *COLT' 98*.

Schlaginhaufen, Andreas and Maryam Kamgarpour (2023). "Identifiability and Generalizability in Constrained Inverse Reinforcement Learning". In: *ICML*.

Schulman, John, Sergey Levine, et al. (2015). "Trust Region Policy Optimization". In: *Proceedings of the 32nd International Conference on Machine Learning*. URL: https://proceedings.mlr.press/v37/schulman15.html.

Schulman, John, Filip Wolski, et al. (July 2017). "Proximal Policy Optimization Algorithms". In.

Searle, Shayle R. (1982). *Matrix Algebra Useful for Statistics*. John Wiley and Sons.

Shapley, Lloyd S. (1953). "Stochastic Games". In: *Proceedings of the National Academy of Sciences* 39, pp. 1095–1100.

Silver, David et al. (Jan. 2016). "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529, pp. 484–489. DOI: 10.1038/nature16961.

Skinner, Burrhus Frederic and Charles Bohris Ferster (1957). *Schedules of Reinforcement*. Appleton-Century-Crofts.

Sutton, Richard S et al. (1999). "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In: *Advances in Neural Information Processing Systems*. Ed. by S. Solla, T. Leen, and K. Müller. Vol. 12.

Wang, Wen et al. (2022). "RL-MD: A Novel Reinforcement Learning Approach for DNA Motif Discovery". In: *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–7.

Williams, R. J. (1992). "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine Learning*, pp. 229–256.

Williams, Ronald J. (1988). *Toward a theory of reinforcement-learning connectionist systems*. Tech. rep. Northeastern University, College of Computer Science.

Ying, Donghao, Yuhao Ding, and Javad Lavaei (2022). "A Dual Approach to Constrained Markov Decision Processes with Entropy Regularization". In: *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. Vol. 151. Proceedings of Machine Learning Research. PMLR.

Zeng, Siliang et al. (2022). "Maximum-Likelihood Inverse Reinforcement Learning with Finite-Time Guarantees". In: *arxiv*.

Ziebart, Brian, J. Andrew Bagnell, and Anind Dey (June 2010). "Modeling Interaction via the Principle of Maximum Causal Entropy". In: pp. 1255–1262.

A

Useful results

This appendix chapter summarizes a few useful generic results that we use in our derivations throughout this document.

A.1. Properties of the log-sum-exp operation

In the following section we discuss the properties of the log-sum-exp function $\log \| \exp \boldsymbol{v} \|_1$ which we very often encounter when deviating expressions related to *entropy-regularized MDP*s.

Observation A.1.1 (Gradient of log-sum-exp is softmax)**.** *The gradient of the log-sum-exp function is the softmax function, indeed:*

$$[\mathit{softmax}(\boldsymbol{v})]_i = \frac{[\exp \boldsymbol{v}]_i}{\sum_j [\exp \boldsymbol{v}]_j} = \frac{[\boldsymbol{v}]_i}{\| \exp \boldsymbol{v} \|_1}, \tag{A.1}$$

is exactly:

$$\nabla(\log \| \exp \boldsymbol{v} \|_1) = \frac{1}{\| \exp \boldsymbol{v} \|_1} \exp \boldsymbol{v}. \tag{A.2}$$

Proposition A.1.1 (Upper-bound on the difference between two log-sum-exp operations)**.** *For any two vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2 \in \mathbb{R}^n$ we have that:*

$$|\log \| \exp \boldsymbol{v}_1 \|_1 - \log \| \exp \boldsymbol{v}_2 \|_1| \leq \| \boldsymbol{v}_1 - \boldsymbol{v}_2 \|_\infty. \tag{A.3}$$

Proof. From the mean-value theorem, we know that there exists a vector $\boldsymbol{v}_c$, which is some convex combination of $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ s.t. the following equality is verified:

$$|\log \| \exp \boldsymbol{v}_1 \|_1 - \log \| \exp \boldsymbol{v}_2 \|_1| = |\langle \boldsymbol{v}_1 - \boldsymbol{v}_2, \nabla \log \| \boldsymbol{v} \|_1 |_{\boldsymbol{v}=\boldsymbol{v}_c} \rangle|, \tag{A.4}$$

and then cleverly using Hölder's inequality we have that:

$$\begin{aligned}
&|\log \| \exp \boldsymbol{v}_1 \|_1 - \log \| \exp \boldsymbol{v}_2 \|_1| && && \text{(A.5)}\\
&= |\langle \boldsymbol{v}_1 - \boldsymbol{v}_2, \nabla \log \| \boldsymbol{v} \|_1 |_{\boldsymbol{v}=\boldsymbol{v}_c} \rangle| && \text{mean value theorem} && \text{(A.6)}\\
&\leq \| \langle \boldsymbol{v}_1 - \boldsymbol{v}_2 \|_\infty \cdot \| \nabla \log \| \boldsymbol{v} \|_1 |_{\boldsymbol{v}=\boldsymbol{v}_c} \|_1 && \text{Hölder's} && \text{(A.7)}\\
&= \| \langle \boldsymbol{v}_1 - \boldsymbol{v}_2 \|_\infty && \| \nabla \log \| \boldsymbol{v} \|_1 |_{\boldsymbol{v}=\boldsymbol{v}_c} \|_1 = \left\| \frac{\exp \boldsymbol{v}_c}{\| \boldsymbol{v}_c \|_1} \right\| = 1. && \text{(A.8)}
\end{aligned}$$

Where the last line is a common observation on log-sum-exp forms. □

B

Omitted proofs and derivations from Chapters 4

This appendix chapter details the results, proofs and derivations omitted from chapter 4.

B.1. Proof of proposition 4.3.5 (Policy-error is upper bounded by Q-value suboptimality)

Proof. To prove the result, we consider some state-action pair $s, a \in S \times A$ and get:

$$|\log \pi^{(t+1)}(a|s) - \log \pi^*_{\tilde{\boldsymbol{r}}^{(t)}}(a|s)| \overset{(i)}{=} \left| \log \left(\frac{\exp Q^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a)}{\sum_{a' \in A} \exp Q^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a')} \right) - \log \left(\frac{\exp Q^{*}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a)}{\sum_{a' \in A} \exp Q^{*}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a')} \right) \right| \tag{B.1}$$

$$\overset{(ii)}{\leq} \left| \log \exp Q^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a) - \log \exp Q^{*}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a) \right| \tag{B.2}$$

$$+ \left| \log \sum_{a' \in A} \exp Q^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a') - \log \sum_{a' \in A} \exp Q^{*}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a'), \right|$$

where (i) holds for the $\pi^{(t+1)}$ term by proposition 2.4.2 (soft-policy iteration) and for term $\pi^*_{\tilde{\boldsymbol{r}}}$ by proposition 2.2.4 (form of the optimal policy), (ii) is just a triangle inequality. Next, using proposition A.1.1 ($|\log \| \exp \boldsymbol{v}_1\|_1 - \log \| \exp \boldsymbol{v}_2\|_1| \leq \|\boldsymbol{v}_1 - \boldsymbol{v}_2\|_\infty$) we get:

$$\left| \log \sum_{a' \in A} \exp Q^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a') - \log \sum_{a' \in A} \exp Q^{*}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a') \right| \leq \max_{a' \in A} \left| Q^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a') - Q^{*}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a') \right|. \tag{B.3}$$

Plugging (B.3) into (B.2), and observing that since these relations hold for any $s, a \in S \times A$ they also hold for the s, a pair where policy error is maximum we have:

$$\| \log \boldsymbol{\pi}^{(t+1)} - \log \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}^{(t)}} \|_\infty \leq 2 \| \boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^{*}_{\tilde{\boldsymbol{r}}^{(t)}} \|_\infty. \tag{B.4}$$

This completes the proof. □

B.2. Proofs of Propositions 4.3.3 and 4.3.4 (Lipschitzness of the Q-function)

We show that Proposition 4.3.3 holds.

Proof. Recall the definition of the Q-value in a *regularized MDP*:

$$Q^{\pi}_{\tilde{\boldsymbol{r}}}(s,a) = (1-\gamma) \mathbb{E}_{\tau \sim \pi} \Big[\sum_{t=0}^{\infty} \gamma^t \big(\tilde{r}(s_t, a_t) + H(\boldsymbol{\pi}(\cdot|s_t)) \big) \Big| s_0 = s, a_0 = a \Big]. \tag{B.5}$$

Taking the difference between the Q values for different diminished rewards and identical policies, we get:

$$\left|Q^{\pi}_{\tilde{\boldsymbol{r}}_1}(s,a) - Q^{\pi}_{\tilde{\boldsymbol{r}}_2}(s,a)\right| \overset{(i)}{=} (1-\gamma)\Bigg|\mathbb{E}_{\tau\sim\pi}\Big[\sum_{t=0}^{\infty}\gamma^t\big(\tilde{r}_1(s_t,a_t) + H(\boldsymbol{\pi}(\cdot|s_t)) \tag{B.6}$$

$$- \tilde{r}_2(s_t,a_t) - H(\boldsymbol{\pi}(\cdot|s_t))\big)\Big|s_0 = s, a_0 = a\Big]\Bigg|$$

$$\overset{(ii)}{\leq} (1-\gamma)\mathbb{E}_{\tau\sim\pi}\Big[\sum_{t=0}^{\infty}\gamma^t\big(|\tilde{r}_1(s_t,a_t) - \tilde{r}_2(s_t,a_t)|\big)\Big|s_0 = s, a_0 = a\Big] \tag{B.7}$$

$$\overset{(iii)}{\leq} (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\big(\|\Phi\|\cdot\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2 + \|\Psi\|\cdot\|\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2\|_2\big) \tag{B.8}$$

$$\overset{(ib)}{=} \|\Phi\|\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2 + \|\Psi\|\|\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2\|_2. \tag{B.9}$$

In (i) we plug in the definition of the regularized Q-values, in (ii) we use Jensen's inequality and finally in (iii) we use that the diminished reward function itself is Lipschitz, finally in (iv) we just use a geometric sum. □

Next, we move on to the proof of Proposition 4.3.4.

Proof. The proposition holds as a corollary of Proposition 4.3.3, to see why observe that:

$$\|\boldsymbol{Q}^*_{\tilde{r}^2} - \boldsymbol{Q}^*_{\tilde{r}^2}\|_\infty = \|\max_{\boldsymbol{\pi}^1\in\Delta_A^S}\boldsymbol{Q}^{\boldsymbol{\pi}^1}_{\tilde{r}^2} - \max_{\boldsymbol{\pi}^2\in\Delta_A^S}\boldsymbol{Q}^{\boldsymbol{\pi}^2}_{\tilde{r}^2}\|_\infty \tag{B.10}$$

$$\leq \max_{\boldsymbol{\pi}\in\Delta_A^S}\|\boldsymbol{Q}^{\boldsymbol{\pi}}_{\tilde{r}^2} - \boldsymbol{Q}^{\boldsymbol{\pi}}_{\tilde{r}^2}\|_\infty \tag{B.11}$$

$$\overset{(i}{\leq} \|\Psi\|\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2 + \|\Psi\|\|\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2\|_2 \tag{B.12}$$

$$\leq C_z\|\boldsymbol{z}_1 - \boldsymbol{z}_2\|, \tag{B.13}$$

where (i) wholds by Proposition 4.3.3. □

B.3. Proof of proposition 4.3.6 (Occupancy measure is Lipschitz with respect to the policies)

Proof. We start by writing out the left-hand-side explicitly:

$$\|\boldsymbol{\mu} - \bar{\boldsymbol{\mu}}\|_\infty = \max_{s,a\in S\times A}|\mu(s,a) - \bar{\mu}(s,a)| \tag{B.14}$$

$$\overset{(i)}{=} \max_{s,a\in S\times A}|\mu_S(s)\pi(a|s) - \bar{\mu}_S(s)\bar{\pi}(a|s)| \tag{B.15}$$

$$\overset{(ii)}{=} \max_{s,a\in S\times A}|\mu_S(s)(\pi(a|s) - \bar{\pi}(a|s)) + \bar{\pi}(a|s)(\mu_S(s) - \bar{\mu}_S(s))| \tag{B.16}$$

$$\overset{(iii)}{\leq} \max_{s,a\in S\times A}\mu_S(s)|(\pi(a|s) - \bar{\pi}(a|s))| + \max_{s,a\in S\times A}\bar{\pi}(a|s)|(\mu_S(s) - \bar{\mu}_S(s))| \tag{B.17}$$

$$\overset{(iv)}{\leq} \|\boldsymbol{\pi} - \bar{\boldsymbol{\pi}}\|_\infty + \|\boldsymbol{\mu}_S(s) - \bar{\boldsymbol{\mu}}_S\|_\infty. \tag{B.18}$$

In (i) we plug in the definition of the state-occupancy measure, in (ii) we rearrange and isolate two difference terms (which are respectively function of the state-occupancy measures and of the policies), in (iii) we just use a triangle inequality which leads us in (iv) to identify $\|\cdot\|_\infty$ norms. Next we thus carry about upper-bounding the $\|\boldsymbol{\mu}_S(s) - \bar{\boldsymbol{\mu}}_S\|_\infty$ term, to do so we will make use of a useful bound on the spectral norm of the sum of two inverse matrices:

$$\|A^{-1} + B^{-1}\| \overset{(i)}{=} \|A^{-1}(A+B)B^{-1}\| \tag{B.19}$$

$$\overset{(ii)}{\leq} \|A^{-1}\| \cdot \|(A+B)\| \cdot \|B^{-1}\| \tag{B.20}$$

$$\overset{(iii)}{=} \frac{\|(A+B)\|}{\sigma_{\min}(A) \cdot \sigma_{\min}(B)}. \tag{B.21}$$

Where (i) holds by the equality $A^{-1} + B^{-1} = A^{-1}(A+B)B^{-1}$ which holds for any two invertible matrices, (ii) holds by submultiplicativity of the spectral norm and (iii) uses the definition of the spectral norm ($\sigma_{\min}(A)$ denotes the minimum eigenvalue of the matrix A). Back to upper-bounding the $\|\boldsymbol{\mu}_S(s) - \bar{\boldsymbol{\mu}}_S\|_\infty$ term, we have:

$$\|\boldsymbol{\mu}_S(s) - \bar{\boldsymbol{\mu}}_S\|_\infty \overset{(i)}{\leq} \|\boldsymbol{\mu}_S(s) - \bar{\boldsymbol{\mu}}_S\|_2 \tag{B.22}$$

$$\overset{(ii)}{=} (1-\gamma)\left\| \left[\left(I - \gamma P^{\boldsymbol{\pi}}\right)^{-1} - \left(I - \gamma P^{\bar{\boldsymbol{\pi}}}\right)^{-1} \right] \boldsymbol{\nu} \right\|_2 \tag{B.23}$$

$$\overset{(iii)}{\leq} (1-\gamma)\left\| \left[\left(I - \gamma P^{\boldsymbol{\pi}}\right)^{-1} - \left(I - \gamma P^{\bar{\boldsymbol{\pi}}}\right)^{-1} \right] \right\| \cdot \|\boldsymbol{\nu}\|_2 \tag{B.24}$$

$$\overset{(iv)}{\leq} (1-\gamma)\left\| \left[\left(I - \gamma P^{\boldsymbol{\pi}}\right)^{-1} - \left(I - \gamma P^{\bar{\boldsymbol{\pi}}}\right)^{-1} \right] \right\| \tag{B.25}$$

$$\overset{(v)}{\leq} (1-\gamma)^{-1}\left\| \left[\left(I - \gamma P^{\boldsymbol{\pi}}\right) - \left(I - \gamma P^{\bar{\boldsymbol{\pi}}}\right) \right] \right\| \tag{B.26}$$

$$= \frac{\gamma}{1-\gamma}\left\|P^{\boldsymbol{\pi}} - P^{\bar{\boldsymbol{\pi}}}\right\|. \tag{B.27}$$

In (i) we just use that the $\|\cdot\|_2$ norm upper-bounds the $\|\cdot\|_\infty$ norm, in (ii) we plug in the closed-form computation of the occupancy measure (which uses the close-loop transition kernel associated with both policies). In (iii) we use the definition of the spectral norm to pull-out the initial distribution term $\boldsymbol{\nu}$ which (iv) we know is less than 1 because it is the $\|\cdot\|_2$ norm of a distribution. Finally in (v) we use the relation (B.21). We are left with the spectral norms of the difference between close-loop transition kernels of both policies. To complete our proof we have to show that this scales linearly with $\|\boldsymbol{\pi} - \bar{\boldsymbol{\pi}}\|_\infty$. Now we just need to bound $\left\|P^{\boldsymbol{\pi}} - P^{\bar{\boldsymbol{\pi}}}\right\|$ which we do as follows:

$$\begin{aligned}
\left\|P^{\boldsymbol{\pi}} - P^{\bar{\boldsymbol{\pi}}}\right\| &\overset{(i)}{\leq} \left\|P^{\boldsymbol{\pi}} - P^{\bar{\boldsymbol{\pi}}}\right\|_F \\
&\overset{(ii)}{=} \sqrt{\sum_{s,s' \in S \times S} \left(P^{\boldsymbol{\pi}}(s'|s) - P^{\bar{\boldsymbol{\pi}}}(s'|s)\right)^2} \\
&\overset{(iii)}{=} \sqrt{\sum_{s,s' \in S \times S} \Big(\sum_{a \in A} P(s'|s,a)(\pi(a|s) - \bar{\pi}(a|s)) \Big)^2} \\
&= \sqrt{\sum_{s,a,s' \in S \times A \times S} P(s'|s,a)^2 (\pi(a|s) - \bar{\pi}(a|s))^2} \\
&= \sqrt{\sum_{s,a \in S \times A} \Big(\sum_{s' \in S} P(s'|s,a)^2 \Big) (\pi(a|s) - \bar{\pi}(a|s))^2} \\
&\leq \sqrt{\sum_{s,a \in S \times A} (\pi(a|s) - \bar{\pi}(a|s))^2} = \|\boldsymbol{\pi} - \bar{\boldsymbol{\pi}}\|_2 \leq \sqrt{nm}\|\boldsymbol{\pi} - \bar{\boldsymbol{\pi}}\|_\infty.
\end{aligned}$$

Where (i) comes from the fact that the Frobenius norm upper bounds the spectral norm, (ii) is by the definition of the Frobenius norm, and (iii) just plugs in the definition of the closed loop transition kernel from there we can just rearange and isolate the $\left(\sum_{s' \in S} P(s'|s,a)^2\right)$ term, which since $P(\cdot, s, a) \in \Delta_S$ we know is less than 1. From there we just observe that we have gotten to the definition of the $l2$ norm. Finally we upper-bound the $\|\cdot\|_2$ norm with the $\|\cdot\|_\infty$ norm.

Putting everything back together we have:

$$\begin{aligned}\|\boldsymbol{\mu}^{\pi} - \boldsymbol{\mu}^{\bar{\pi}}\|_2 &\leq \|\boldsymbol{\pi} - \bar{\boldsymbol{\pi}}\|_2 + \|\boldsymbol{\mu}_s - \bar{\boldsymbol{\mu}}_s\|_\infty \\ &\leq \|\boldsymbol{\pi} - \bar{\boldsymbol{\pi}}\|_2 + \frac{\gamma}{1-\gamma}\left\|P^{\boldsymbol{\pi}} - P^{\bar{\boldsymbol{\pi}}}\right\| \\ &\leq \|\boldsymbol{\pi} - \bar{\boldsymbol{\pi}}\|_2 + \frac{\sqrt{nm}\gamma}{1-\gamma}\|\boldsymbol{\pi} - \bar{\boldsymbol{\pi}}\|_2 \\ &= \frac{1 + (\sqrt{nm} - 1)\gamma}{1-\gamma}\|\boldsymbol{\pi} - \bar{\boldsymbol{\pi}}\|_\infty.\end{aligned}$$

□

B.4. Proof of lemma 4.3.2 (Constraint violation)

Proof. We start by considering the constraint violation directly, we have:

$$\|[\boldsymbol{b} - \boldsymbol{K}(\boldsymbol{\theta}^{(*)})]_+\|_\infty = \|[\boldsymbol{b} - \boldsymbol{K}^* + \boldsymbol{K}^* - \boldsymbol{K}(\boldsymbol{\theta}^{(*)})]_+\|_\infty \tag{B.28}$$

$$\overset{(i)}{\leq} \|[\boldsymbol{b} - \boldsymbol{K}^*]_+\|_\infty + \|[\boldsymbol{K}^* - \boldsymbol{K}(\boldsymbol{\theta}^{(*)})]_+\|_\infty \tag{B.29}$$

$$\overset{(ii)}{\leq} \|[\boldsymbol{K}^* - \boldsymbol{K}(\boldsymbol{\theta}^{(*)})]_+\|_\infty. \tag{B.30}$$

Where in (i) we use that $\max\{a+b, 0\} < \max\{a, 0\} + \max\{b, 0\}$ and a triangle inequality and in (2) we use that $\boldsymbol{b} - \boldsymbol{K}^*$ is the 0 vector (all constraints are satisfied in the optimal solution). Next we look at the difference of cost vector:

$$\|[\boldsymbol{K}^* - \boldsymbol{K}(\boldsymbol{\theta}^{(*)})]_+\|_\infty = \|\boldsymbol{K}^* - \boldsymbol{K}(\boldsymbol{\theta}^{(*)})\|_\infty = \|\Psi^\top(\boldsymbol{\mu}^{(*)} - \boldsymbol{\mu}^*)\|_\infty \tag{B.31}$$

$$\overset{(i)}{\leq} \|\Psi^\top\|_\infty \cdot \|(\boldsymbol{\mu}^{(*)} - \boldsymbol{\mu}^*)\|_\infty \tag{B.32}$$

$$\overset{(ii)}{\leq} \sqrt{d}\|\Psi\|\frac{1 + \gamma\sqrt{nm}}{1-\gamma}\|\boldsymbol{\pi} - \boldsymbol{\pi}^*\|_\infty \tag{B.33}$$

$$\overset{(iii)}{\leq} \sqrt{d}\|\Psi\|\frac{1 + \gamma\sqrt{nm}}{1-\gamma}\|\log \boldsymbol{\pi} - \log \boldsymbol{\pi}^*\|_\infty \tag{B.34}$$

$$\overset{(iv)}{\leq} \sqrt{d}\|\Psi\|\frac{1 + \gamma\sqrt{nm}}{1-\gamma}\left(4\|\phi\|_1\gamma^T + 4C_z'\frac{1-\gamma^T}{1-\gamma}\frac{1}{T^u}\right). \tag{B.35}$$

In (i) we use the $\|\cdot\|_\infty$ operator norm on the matrix $\Psi^\top$ and then in (ii) we upper bound the $\|\cdot\|_\infty$ operator norm by the spectral norm, and use proposition 4.3.6, in (iii) we use that $|x - y| < |\log x - \log y|$ when $x, y \in (0, 1)$ and finally in (iv) we plug in the convergence rate from our local convergence Lemma (Lemma 4.3.3). The proof is complete. □

C

Omitted proofs and derivations from Chapters 5

This appendix chapter details the results, proofs and derivations omitted from chapter 5.

C.1. Proof of Proposition 5.2.2 (Pertubed policy step)

Proof. This proof is very similar to the one provided in Appendix C.6 of [Cen et al. 2021], with a few key differences that we highlight. Before we start we state the definition of the advantage function:

$$A_{\boldsymbol{r}}^{\boldsymbol{\pi_\theta}}(s,a) = Q_{\boldsymbol{r}}^{\boldsymbol{\pi_\theta}}(s,a) - \beta \log \pi_{\boldsymbol{\theta}}(a|s) - Q_{\boldsymbol{r}}^{\boldsymbol{\pi_\theta}}(s). \tag{C.1}$$

The gradient $\nabla_\theta L(\boldsymbol{\theta}, \boldsymbol{r}, \boldsymbol{\lambda})$ is given by:

$$\begin{aligned}
\nabla_\theta L(\boldsymbol{\theta}, \boldsymbol{r}, \boldsymbol{\lambda}) &= \nabla_\theta \Big[J(\boldsymbol{\pi_\theta}, \boldsymbol{r}) - J(\boldsymbol{\pi}^E, \boldsymbol{r}) + \langle \boldsymbol{\lambda}, \boldsymbol{b} - \Psi^\top \boldsymbol{\mu}^{\boldsymbol{\pi_\theta}} \rangle \Big] && \text{(C.2)} \\
&= \nabla_\theta J(\boldsymbol{\pi_\theta}, \boldsymbol{r}) - \nabla_\theta J(\boldsymbol{\pi}^E, \boldsymbol{r}) + \nabla_\theta \langle \boldsymbol{\lambda}, \boldsymbol{b} - \Psi^\top \boldsymbol{\mu}^{\boldsymbol{\pi_\theta}} \rangle && \text{(C.3)} \\
&= \nabla_\theta J(\boldsymbol{\pi_\theta}, \tilde{\boldsymbol{r}}) - \nabla_\theta J(\boldsymbol{\pi}^E, \tilde{\boldsymbol{r}}) && \text{(C.4)} \\
&= \nabla_\theta J(\boldsymbol{\pi_\theta}, \tilde{\boldsymbol{r}}). && \text{(C.5)}
\end{aligned}$$

In the entropy regularized setting, the gradient of the return is given by:

$$\frac{\partial J(\boldsymbol{\pi_\theta}, \tilde{\boldsymbol{r}})}{\partial \theta(s,a)} = \mu_{\boldsymbol{\pi}^{\boldsymbol{\theta}}}(s,a) A_{\tilde{\boldsymbol{r}}}^{\boldsymbol{\pi_\theta}}(s,a), \tag{C.6}$$

for a detailed derivation of (C.6) we refer to [Cen et al. 2021]. Recall that the NPG step is given by (in the exact gradient setting):

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} + \eta_\theta \big(\mathfrak{F}^{\boldsymbol{\theta}}\big)^\dagger \nabla_\theta L(\boldsymbol{\theta}, \boldsymbol{r}, \boldsymbol{\lambda}) = \boldsymbol{\theta}^{(t)} + \eta_\theta \big(\mathfrak{F}^{\boldsymbol{\theta}}\big)^\dagger \nabla_\theta J(\boldsymbol{\pi_\theta}, \tilde{\boldsymbol{r}}), \tag{C.7}$$

introducing the stochastic oracle gradients we have:

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} + \eta_\theta \big(\mathfrak{F}^{\boldsymbol{\theta}}\big)^\dagger \boldsymbol{g}_{\boldsymbol{\theta}}. \tag{C.8}$$

One key observation is that the step $\boldsymbol{w}_{\boldsymbol{\theta}} = \big(\mathfrak{F}^{\boldsymbol{\theta}}\big)^\dagger \boldsymbol{g}_{\boldsymbol{\theta}}$ is given by:

$$\boldsymbol{w}_{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{w} \in \mathbb{R}^{nm}} \big\| \mathfrak{F}^{\boldsymbol{\theta}} \boldsymbol{w} - \boldsymbol{g}_{\boldsymbol{\theta}} \big\|_2^2. \tag{C.9}$$

Consider the matrix vector product of the decision variable $\boldsymbol{w}$ with the Fisher information matrix (Definition 2.4.2):

$$\mathfrak{F}^{\boldsymbol{\theta}} \boldsymbol{w} = \mathbb{E}_{s,a \sim \boldsymbol{\mu}_{\boldsymbol{\pi_\theta}}} \Big[(\nabla_{\boldsymbol{\theta}} \log \pi(a|s)) (\nabla_{\boldsymbol{\theta}} \log \pi(a|s))^\top \boldsymbol{w} \Big] \tag{C.10}$$

$$\overset{(i)}{=} \sum_{s,a} \mu_{\boldsymbol{\pi_\theta}}(s,a)\big(w(s,a) - c(s)\big). \tag{C.11}$$

Where $c(s) = \sum_{s,a} \pi_{\boldsymbol{\theta}}(s,a) w(s,a)$ again we refer the reader to [Cen et al. 2021] for a detailed derivation of step (i). Now we are recall Assumption 5.2.1:

$$\mathbb{E}\Big[\|\boldsymbol{g_\theta} - \nabla_\theta L\|\Big] \leq \delta. \tag{C.12}$$

We define $\boldsymbol{\sigma}_\theta = \boldsymbol{g_\theta} - \nabla_\theta L$ and note that by the $\|\cdot\|_\infty \leq \|\cdot\|_2$ the assumption implies that for any $(s,a) \in S \times A$:

$$\mathbb{E}\left[\left|g_{\boldsymbol{\theta}}(s,a) - \frac{\partial J(\boldsymbol{\pi_\theta}, \tilde{\boldsymbol{r}})}{\partial \theta(s,a)}\right|\right] \leq \mathbb{E}\Big[\big\|\boldsymbol{\sigma_\theta}\big\|_\infty\Big] = \mathbb{E}\Big[\|\boldsymbol{g_\theta} - \nabla_\theta L\|_\infty\Big] \leq \delta. \tag{C.13}$$

Using expression (C.11) and observation (C.9) we are now equipped to compute the NPG step:

$$\big\|\mathfrak{F}^{\boldsymbol{\theta}}\boldsymbol{w} - \boldsymbol{g_\theta}\big\|_2^2 = \sum_{s,a}\Big(\mu_{\boldsymbol{\pi_\theta}}(s,a)\big(w(s,a) - c(s)\big) - g_{\boldsymbol{\theta}(s,a)}\Big)^2 \tag{C.14}$$

$$= \sum_{s,a}\Big(\mu_{\boldsymbol{\pi_\theta}}(s,a)\big(w(s,a) - c(s)\big) - \mu_{\boldsymbol{\pi^\pi}}(s,a) A^{\boldsymbol{\pi_\theta}}_{\tilde{\boldsymbol{r}}}(s,a) + \sigma_{\boldsymbol{\sigma}}(s,a)\Big)^2 \tag{C.15}$$

$$= \sum_{s,a}\left(\mu_{\boldsymbol{\pi_\theta}}(s,a)\Big(w(s,a) - c(s) - A^{\boldsymbol{\pi_\theta}}_{\tilde{\boldsymbol{r}}}(s,a) + \frac{\sigma_{\boldsymbol{\sigma}}(s,a)}{\mu_{\boldsymbol{\pi_\theta}}(s,a)}\Big)\right)^2. \tag{C.16}$$

It is easy to see that the solution $\boldsymbol{w_\theta}$ is given by:

$$w(s,a) = c(s) + A^{\boldsymbol{\pi_\theta}}_{\tilde{\boldsymbol{r}}}(s,a) - \frac{\sigma_{\boldsymbol{\sigma}}(s,a)}{\mu_{\boldsymbol{\pi_\theta}}(s,a)}, \tag{C.17}$$

and that by Assumption 5.2.1 the approximate gradient step in a neighborhood of the exact gradient step (which is just given by the $\|\boldsymbol{\sigma_\theta}\|_2 = 0$):

$$\mathbb{E}\left[\Big\|\boldsymbol{w_\theta} - \big(\mathfrak{F}^{\boldsymbol{\theta}}\big)^\dagger \nabla_\theta J(\boldsymbol{\pi_\theta}, \tilde{\boldsymbol{r}})\Big\|_\infty\right] = \mathbb{E}\left[\Big\|\big(\mathfrak{F}^{\boldsymbol{\theta}}\big)^\dagger \boldsymbol{g_\theta} - \big(\mathfrak{F}^{\boldsymbol{\theta}}\big)^\dagger \nabla_\theta J(\boldsymbol{\pi_\theta}, \tilde{\boldsymbol{r}})\Big\|_\infty\right] \tag{C.18}$$

$$\leq \Big|\frac{\delta}{\mu_{\mathsf{min}}}\Big|. \tag{C.19}$$

So far, we have established that:

$$\Big[\big(\mathfrak{F}^{\boldsymbol{\theta}}\big)^\dagger \boldsymbol{g_\theta}\Big](s,a) = A^{\boldsymbol{\pi_\theta}}_{\tilde{\boldsymbol{r}}}(s,a) + c(s) - \frac{\sigma_{\boldsymbol{\sigma}}(s,a)}{\mu_{\boldsymbol{\pi_\theta}}(s,a)}, \tag{C.20}$$

we will now show how this translates to the policy step. We will abuse notation and write $\boldsymbol{\pi}^{(t)} = \boldsymbol{\pi}_{\boldsymbol{\theta}^{(t)}}$ have that:

$$\pi^{(t+1)}(a|s) \overset{(i)}{\propto} \mathsf{exp}\big(\theta^{(t+1)}(s,a)\big) = \mathsf{exp}\big(\theta^{(t)}(s,a) + \eta_\theta\Big[\big(\mathfrak{F}^{\boldsymbol{\theta}}\big)^\dagger \boldsymbol{g_\theta}\Big](s,a)\big) \tag{C.21}$$

$$\overset{(ii)}{\propto} \mathsf{exp}\left(\theta^{(t)}(s,a) + \eta_\theta A^{\boldsymbol{\pi_\theta}}_{\tilde{\boldsymbol{r}}}(s,a) - \frac{\sigma_{\boldsymbol{\sigma}}(s,a)}{\mu_{\boldsymbol{\pi_\theta}}(s,a)}\right) \tag{C.22}$$

$$\overset{(iii)}{\propto} \pi^{(t)}(a|s)\,\mathsf{exp}\left(\eta_\theta Q^{\boldsymbol{\pi_\theta}}_{\tilde{\boldsymbol{r}}}(s,a) - \beta\,\mathsf{log}\,\pi^{(t)}(a|s) - \frac{\sigma_{\boldsymbol{\sigma}}(s,a)}{\mu_{\boldsymbol{\pi_\theta}}(s,a)}\right) \tag{C.23}$$

$$\overset{(iv)}{\propto} \big(\pi^{(t)}(a|s)\big)^{1-\eta_\theta\beta}\,\mathsf{exp}\left(\eta_\theta Q^{\boldsymbol{\pi_\theta}}_{\tilde{\boldsymbol{r}}}(s,a) - \frac{\sigma_{\boldsymbol{\sigma}}(s,a)}{\mu_{\boldsymbol{\pi_\theta}}(s,a)}\right) \tag{C.24}$$

$$= \big(\pi^{(t)}(a|s)\big)^{1-\eta_\theta\beta}\,\mathsf{exp}\Big(\eta_\theta \hat{Q}^{\boldsymbol{\pi_\theta}}_{\tilde{\boldsymbol{r}}}(s,a)\Big). \tag{C.25}$$

Where (i) is obtained from the softmax parameterization (Assumption 4.2.1) and (ii) from the NPG step, we neglect $c(s)$ because the softmax normalizes away all terms which have the same value for all elements across s. In (iii) we plug in the definition of the advantage (again the V term gets normalized away by softmax). Finally in step (iv) we factor all policy-related terms in from of the exponential. Finally, we identify $\hat{\boldsymbol{Q}}$ which satisfies $\big\|\hat{\boldsymbol{Q}}_r^{(t)} - \boldsymbol{Q}_r^{(t)}\big\|_\infty \leq \breve{\delta}$ by definition of $\boldsymbol{\sigma}$. □

C.2. Characterization of the estimators

C.2.1. Proof of Proposition 5.2.1 (Policy gradient estimator)

Proof. To get to the bounded mean euclidean distance error from Proposition 5.2.1 we start by writing out explicitly our exact gradients and our gradient estimator:

$$\boldsymbol{g}_\theta^{(t)} = \frac{1}{B}\sum_{i=1}^{B}\left(\sum_{h=0}^{H-1}\gamma^h\Big(\Sigma_{j=0}^{h}\nabla_\theta \log \pi_{\boldsymbol{\theta}^{(t)}}(a_j^i|s_j^i)\Big)b_h\right), \tag{C.26}$$

$$\nabla_\theta L(\boldsymbol{\theta},\boldsymbol{w},\boldsymbol{\lambda}) = \mathbb{E}\left[\sum_{h=0}^{+\infty}\gamma^h\Big(\Sigma_{j=0}^{h}\nabla_\theta \log \pi_{\boldsymbol{\theta}^{(t)}}(a_j^i|s_j^i)\Big)b_h\right], \tag{C.27}$$

where we write $b_h = (\tilde{r}_{\boldsymbol{\lambda}^{(t)}}(s_h,a_h) - \beta \log \pi_{\boldsymbol{\theta}^{(t)}})$ for conciseness. From (C.26) and (C.27) we get:

$$\mathbb{E}\Big[\|\boldsymbol{g}_\theta^{(t)} - \nabla_\theta L(\boldsymbol{\theta},\boldsymbol{w},\boldsymbol{\lambda})\|_2\Big] = \mathbb{E}\Bigg[\Bigg\|\frac{1}{B}\sum_{i=1}^{B}\left(\sum_{h=0}^{H-1}\gamma^h\Big(\Sigma_{j=0}^{h}\nabla_\theta \log \pi_{\boldsymbol{\theta}^{(t)}}(a_j^i|s_j^i)\Big)b_h\right) \tag{C.28}$$

$$- \mathbb{E}\left[\sum_{h=0}^{+\infty}\gamma^h\Big(\Sigma_{j=0}^{h}\nabla_\theta \log \pi_{\boldsymbol{\theta}^{(t)}}(a_j^i|s_j^i)\Big)b_h\right]\Bigg\|_2\Bigg]$$

$$\overset{(i)}{=} \mathbb{E}\Bigg[\Bigg\|\frac{1}{B}\sum_{i=1}^{B}\left(\sum_{h=0}^{H-1}\gamma^h\Big(\Sigma_{j=0}^{h}\nabla_\theta \log \pi_{\boldsymbol{\theta}^{(t)}}(a_j^i|s_j^i)\Big)b_h\right) \tag{C.29}$$

$$- \mathbb{E}\left[\sum_{h=0}^{H-1}\gamma^h\Big(\Sigma_{j=0}^{h}\nabla_\theta \log \pi_{\boldsymbol{\theta}^{(t)}}(a_j^i|s_j^i)\Big)b_h\right]$$

$$- \mathbb{E}\left[\sum_{h=H}^{+\infty}\gamma^h\Big(\Sigma_{j=0}^{h}\nabla_\theta \log \pi_{\boldsymbol{\theta}^{(t)}}(a_j^i|s_j^i)\Big)b_h\right]\Bigg\|_2\Bigg]$$

$$\overset{(ii)}{\leq} \mathbb{E}\Bigg[\Bigg\|\frac{1}{B}\sum_{i=1}^{B}\left(\sum_{h=0}^{H-1}\gamma^h\Big(\Sigma_{j=0}^{h}\nabla_\theta \log \pi_{\boldsymbol{\theta}^{(t)}}(a_j^i|s_j^i)\Big)b_h\right) \tag{C.30}$$

$$- \mathbb{E}\left[\sum_{h=0}^{H-1}\gamma^h\Big(\Sigma_{j=0}^{h}\nabla_\theta \log \pi_{\boldsymbol{\theta}^{(t)}}(a_j^i|s_j^i)\Big)b_h\right]\Bigg\|_2\Bigg]$$

$$+ \left\|\mathbb{E}\Big[\sum_{h=H}^{+\infty}\gamma^h\Big(\Sigma_{j=0}^{h}\nabla_\theta \log \pi_{\boldsymbol{\theta}^{(t)}}(a_j^i|s_j^i)\Big)b_h\Big]\right\|_2$$

$$= \mathbb{E}\Big[\Big\|\boldsymbol{g}_\theta^{(t)} - \mathbb{E}[\boldsymbol{g}_\theta^{(t)}]\Big\|_2\Big] \tag{C.31}$$

$$+ \left\|\mathbb{E}\Big[\sum_{h=H}^{+\infty}\gamma^h\Big(\Sigma_{j=0}^{h}\nabla_\theta \log \pi_{\boldsymbol{\theta}^{(t)}}(a_j^i|s_j^i)\Big)b_h\Big]\right\|_2$$

In where (i) comes from splitting the infinite sum in two "sections" (terms 0 to $H-1$ and terms H to ∞) and (ii) is a simple triangle inequality. We are left with two terms, a term induced by the truncation and a term akin to variance of the gradient estimator. We will consider both terms separately, starting with the truncation term:

$$\left\|\mathbb{E}\Big[\sum_{h=H}^{+\infty}\gamma^h b_h\Big(\Sigma_{j=0}^{h}\nabla_\theta \log \pi_{\boldsymbol{\theta}^{(t)}}(a_j^i|s_j^i)\Big)\Big]\right\|_2 \overset{(i)}{\leq} \mathbb{E}\left[\Big\|\sum_{h=H}^{+\infty}\gamma^h b_h\Big(\Sigma_{j=0}^{h}\nabla_\theta \log \pi_{\boldsymbol{\theta}^{(t)}}(a_j^i|s_j^i)\Big)\Big\|_2\right] \tag{C.32}$$

$$\overset{(ii)}{\leq} \mathbb{E}\left[\sum_{h=H}^{+\infty}\gamma^h b_h \Sigma_{j=0}^{h}\Big\|\nabla_\theta \log \pi_{\boldsymbol{\theta}^{(t)}}(a_j^i|s_j^i)\Big\|_2\right] \overset{(iii)}{\leq} \mathbb{E}\left[\sum_{h=H}^{+\infty}\gamma^h b_{\max}\Sigma_{j=0}^{h}\Big\|\nabla_\theta \log \pi_{\boldsymbol{\theta}^{(t)}}(a_j^i|s_j^i)\Big\|_2\right] \tag{C.33}$$

$$\overset{(iv)}{\leq} \sqrt{2}b_{\max}\sum_{h=H}^{+\infty}h\gamma^h \leq \sqrt{2}b_{\max}H\gamma^{H-1}\sum_{j=0}^{+\infty}(j)\gamma^j = \sqrt{2}b_{\max}\frac{\gamma^2(2-\gamma)}{(1-\gamma)^2}H\gamma^{H-1} \tag{C.34}$$

Where (i) is an application of Jensen's inequality, (ii) is a simple triangle inequality and (iii) introduces $b_{\max}$, the maximum value that can be taken by b_h. Finally, inequality (iv) uses that when π is softmax parameterized, for any (s,a) we have that $\|\nabla_\theta \log \pi_{\boldsymbol{\theta}^{(t)}}(a_j^i|s_j^i)\|_2^2 \leq 1 + \|\boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s)\|_2 - 2\pi^{(\boldsymbol{\theta})}(s,a) \leq 1 + \|\boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s)\|_1 \leq 2$. We are left with explicitly computing the upper-bound of b_h:

$$b_h \leq b_{\max} = \|\phi\| + \lambda_{\max} + \beta \log \pi_{\min}. \tag{C.35}$$

Now onto the remaining term, first we note that it is upper-bounded by the variance of the GPOMDP estimator,

$$\mathbb{E}\Big[\big\|\boldsymbol{g}_\theta^{(t)} - \mathbb{E}[\boldsymbol{g}_\theta^{(t)}]\big\|_2\Big] = \sqrt{\left(\mathbb{E}\left[\big\|\boldsymbol{g}_\theta^{(t)} - \mathbb{E}[\boldsymbol{g}_\theta^{(t)}]\big\|_2\right]\right)^2} \overset{(i)}{\leq} \sqrt{\mathbb{E}\Big[\big\|\boldsymbol{g}_\theta^{(t)} - \mathbb{E}[\boldsymbol{g}_\theta^{(t)}]\big\|_2^2\Big]} = \sqrt{\mathsf{Var}(\boldsymbol{g}_\theta^{(t)})}, \tag{C.36}$$

where (i) holds by Jensen's inequality and concavity of the square root function. Next we just use Lemma 3.7 from [Y. Ding, Zhang, and Lavaei 2021] which quantifies said variance:

$$\mathbb{E}\Big[\big\|\boldsymbol{g}_\theta^{(t)} - \mathbb{E}[\boldsymbol{g}_\theta^{(t)}]\big\|_2\Big] \leq \frac{24(\|\phi\|_1 + \lambda_{\max} + \beta^2(\log m)^2)}{B^2(1-\gamma)^4}. \tag{C.37}$$

Bringing (C.34) and (C.37) together into (C.32) we find our result:

$$\mathbb{E}\Big[\|\boldsymbol{g}_\theta^{(t)} - \nabla_\theta L(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\lambda})\|_2\Big] \leq \sqrt{2} b_{\max} \frac{\gamma^2(2-\gamma)}{(1-\gamma)^2} H\gamma^{H-1} + \frac{2\sqrt{6}\sqrt{\|\phi\|_1 + \lambda_{\max} + 24\beta^2(\log m)^2}}{B(1-\gamma)^2}, \tag{C.38}$$

□

C.2.2. Proof of Proposition 5.2.2 (Reward gradient estimator)

Proof. Our proof will be quite similar to the one of Proposition 5.2.1 in the sense that we will isolate a truncation term and variance term and then bound both of them, recall that our gradient estimator is given by:

$$\hat{\boldsymbol{\varphi}} - \hat{\boldsymbol{\varphi}}^E = \frac{1-\gamma}{B} \sum_{i=1}^{B} \Big(\sum_{h=0}^{H-1} \gamma^h \boldsymbol{\phi}(s_h^i, a_h^i) \Big) - \hat{\boldsymbol{\varphi}}^E, \tag{C.39}$$

while the exact gradient is given by:

$$\boldsymbol{\varphi} - \boldsymbol{\varphi}^E = \mathbb{E}\left[(1-\gamma) \Big(\sum_{h=0}^{+\infty} \gamma^h \boldsymbol{\phi}(s_h^i, a_h^i) \Big) - \hat{\boldsymbol{\varphi}}^E \right]. \tag{C.40}$$

Inserting these expressions into the left-hand side of (5.20), we get:

$$\mathbb{E}\left[\left\| \frac{1-\gamma}{B} \sum_{i=1}^{B} \Big(\sum_{h=0}^{H-1} \gamma^h \boldsymbol{\phi}(s_h^i, a_h^i) \Big) - \mathbb{E}\Big[(1-\gamma) \sum_{h=0}^{+\infty} \gamma^h \boldsymbol{\phi}(s_h^i, a_h^i) \Big] \right\|_2 \right] \tag{C.41}$$

$$\overset{(i)}{\leq} \mathbb{E}\left[\left\| \frac{1-\gamma}{B} \sum_{i=1}^{B} \Big(\sum_{h=0}^{H-1} \gamma^h \boldsymbol{\phi}(s_h^i, a_h^i) \Big) - \mathbb{E}\Big[(1-\gamma) \sum_{h=0}^{H-1} \gamma^h \boldsymbol{\phi}(s_h^i, a_h^i) \Big] \right\|_2 \right] + \tag{C.42}$$

$$\left\| \mathbb{E}\Big[(1-\gamma) \sum_{h=H}^{\infty} \gamma^h \boldsymbol{\phi}(s_h^i, a_h^i) \Big] \right\|_2$$

$$= \mathbb{E}\Big[\big\| \boldsymbol{g}_{\boldsymbol{w}}^{(t)} - \mathbb{E}[\boldsymbol{g}_{\boldsymbol{w}}^{(t)}] \big\|_2 \Big] + \left\| \mathbb{E}\Big[(1-\gamma) \sum_{h=H}^{\infty} \gamma^h \boldsymbol{\phi}(s_h^i, a_h^i) \Big] \right\|_2 \tag{C.43}$$

In step (i) we simply split the terms of the infinite of (C.40) into the first T terms and the tail of the sum. Taking a triangle inequality we are left two terms one associated with truncation and one associated with variance, we start by considering the truncation term:

$$\left\| \mathbb{E}\Big[(1-\gamma) \sum_{h=H}^{\infty} \gamma^h \boldsymbol{\phi}(s_h^i, a_h^i) \Big] \right\|_2 \leq \mathbb{E}\Big[(1-\gamma) \big\| \sum_{h=H}^{\infty} \gamma^h \boldsymbol{\phi}(s_h^i, a_h^i) \big\|_2 \Big] \tag{C.44}$$

$$\leq \mathbb{E}\Big[(1-\gamma)\sum_{h=H}^{\infty}\big\|\gamma^h\boldsymbol{\phi}(s_h^i,a_h^i)\big\|_2 \leq \mathbb{E}\Big[(1-\gamma)\sum_{h=H}^{\infty}\gamma^h\phi_{\mathsf{max}}\Big] \tag{C.45}$$

$$\leq \boldsymbol{\phi}_{\mathsf{max}}\gamma^H(1-\gamma)\sum_{i=0}^{\infty}\gamma^i = \gamma^H\phi_{\mathsf{max}}. \tag{C.46}$$

Where ϕ_{max} is the maximum $\|\cdot\|_2$ norm of any column of the feature matrix ϕ. We are now ready to move on to the variance term, we have:

$$\mathbb{E}\Big[\Big\|\boldsymbol{g}_{\boldsymbol{w}}^{(t)} - \mathbb{E}[\boldsymbol{g}_{\boldsymbol{w}}^{(t)}]\Big\|_2\Big] = \mathbb{E}\left[\sqrt{\Big\|\boldsymbol{g}_{\boldsymbol{w}}^{(t)} - \mathbb{E}[\boldsymbol{g}_{\boldsymbol{w}}^{(t)}]\Big\|_2^2}\right] \overset{(i)}{\leq} \sqrt{\mathbb{E}\left[\Big\|\boldsymbol{g}_{\boldsymbol{w}}^{(t)} - \mathbb{E}[\boldsymbol{g}_{\boldsymbol{w}}^{(t)}]\Big\|_2^2\right]} = \sqrt{\mathsf{Var}(\boldsymbol{g}_{\boldsymbol{w}}^{(t)})}. \tag{C.47}$$

Here (i) holds by Jensen's inequality. We are just left with finding a bound for the variance of our estimator, which we do by considering the single element batch first, we have:

$$\mathbb{E}\Big[\big\|\boldsymbol{g}_{\boldsymbol{w}} - \mathbb{E}\big[\boldsymbol{g}_{\boldsymbol{w}}\big]\big\|_2^2\Big] = (1-\gamma)^2\mathbb{E}\left[\left\|\sum_{h=0}^{H-1}\gamma^h\boldsymbol{\phi}(s_h^i,a_h^i) - \mathbb{E}\Big[\sum_{h=0}^{H-1}\gamma^h\boldsymbol{\phi}(s_h^i,a_h^i)\Big]\right\|_2^2\right] \tag{C.48}$$

$$= (1-\gamma)^2\mathbb{E}\left[\left\|\sum_{h=0}^{H-1}\gamma^h\Big[\boldsymbol{\phi}(s_h^i,a_h^i) - \mathbb{E}\big[\boldsymbol{\phi}(s_h^i,a_h^i)\big]\Big]\right\|_2^2\right] \tag{C.49}$$

$$\leq (1-\gamma)^2\mathbb{E}\left[\sum_{h=0}^{H-1}\gamma^h\big\|\boldsymbol{\phi}(s_h^i,a_h^i) - \mathbb{E}\big[\boldsymbol{\phi}(s_h^i,a_h^i)\big]\big\|_2^2\right] \tag{C.50}$$

$$\leq (1-\gamma)^2\sum_{h=0}^{H-1}\gamma^h 4\phi_{\mathsf{max}}^2 = (1-\gamma)^2 4\phi_{\mathsf{max}}^2\sum_{h=0}^{H-1} = 4(1-\gamma)\phi_{\mathsf{max}}^2. \tag{C.51}$$

Now since we take batches with B independent samples this gives a variance of $\frac{4(1-\gamma)\phi_{\mathsf{max}}^2}{B^2}$ hence putting everything back together we get:

$$\mathbb{E}\Big[\|\boldsymbol{g}_{\boldsymbol{w}}^{(t)} - \nabla_{\boldsymbol{w}}L(\boldsymbol{\theta},\boldsymbol{w},\boldsymbol{\lambda})\|_2\Big] \leq \phi_{\mathsf{max}}\left(\frac{2\sqrt{1-\gamma}}{B} + \gamma^H\right). \tag{C.52}$$

□

C.3. Local convergence bounds

C.3.1. Proof of Proposition 5.2.5 (First auxiliary sequence term)

Proof. We first consider Proposition 5.2.5, to do so, we start with the recursion defined in 5.38. Consider the vector $\boldsymbol{Q}^*_{\tilde{r}(t+1)} - \beta\log\boldsymbol{\xi}^{(t+1)}$:

$$\boldsymbol{Q}^*_{\tilde{r}(t+1)} - \beta\log\boldsymbol{\xi}^{(t+1)} \overset{(i)}{=} \boldsymbol{Q}^*_{\tilde{r}(t+1)} - \alpha\beta\log\boldsymbol{\xi}^{(t)} - (1-\alpha)\hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}(t)} \tag{C.53}$$

$$\overset{(ii)}{=} \alpha\big(\boldsymbol{Q}^*_{\tilde{r}(t)} - \beta\log\boldsymbol{\xi}^{(t)}\big) + (1-\alpha)\big(\boldsymbol{Q}^*_{\tilde{r}(t)} - \boldsymbol{Q}^{(t)}_{\tilde{r}(t)}\big) \tag{C.54}$$

$$+ (1-\alpha)\big(\boldsymbol{Q}^{(t)}_{\tilde{r}(t)} - \hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}(t)}\big) - \big(\boldsymbol{Q}^*_{\tilde{r}(t+1)} - \boldsymbol{Q}^*_{\tilde{r}(t)}\big).$$

Where in (i) we plug in the update rule for the auxiliary sequence as defined in (5.38), and in (ii) we reorganize the equation to isolate quantities that are relevant to our study[1]. Taking $\|\cdot\|_\infty$ norm and using the triangle inequality we get:

$$x_2^{(t+1)} = \|\boldsymbol{Q}^*_{\tilde{r}(t+1)} - \beta\log\boldsymbol{\xi}^{(t+1)}\|_\infty \leq \alpha\|\boldsymbol{Q}^*_{\tilde{r}(t)} - \beta\log\boldsymbol{\xi}^{(t)}\|_\infty + (1-\alpha)\|\boldsymbol{Q}^*_{\tilde{r}(t)} - \boldsymbol{Q}^{(t)}_{\tilde{r}(t)}\|_\infty \tag{C.55}$$

$$+ (1-\alpha)\|\boldsymbol{Q}^{(t)}_{\tilde{r}(t)} - \hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}(t)}\|_\infty + \|\boldsymbol{Q}^*_{\tilde{r}(t+1)} - \boldsymbol{Q}^*_{\tilde{r}(t)}\|_\infty$$

$$\overset{(i)}{\leq} \alpha x_2^{(t)} + (1-\alpha)x_1^{(t)} \tag{C.56}$$

[1] It is easily verified that line (C.66) simplifies back into (C.65)

$$+ (1-\alpha)\|\boldsymbol{Q}^{(t)}_{\tilde{r}(t)} - \hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}(t)}\|_\infty + \|\boldsymbol{Q}^{*}_{\tilde{r}(t+1)} - \boldsymbol{Q}^{*}_{\tilde{r}(t)}\|_\infty.$$

In (i) we just plug in the scalar error terms from their definitions (5.40) and (5.41). We now take expectations (over the algorithm step) on both sides to get:

$$\begin{aligned}
\mathbb{E}_{(t+1)}\big[x_2^{(t+1)}\big] &\leq (1-\alpha)\mathbb{E}_{(t+1)}\big[x_1^{(t)}\big] + \alpha\mathbb{E}_{(t+1)}\big[x_2^{(t)}\big] + (1-\alpha)\mathbb{E}_{(t+1)}\big[\|\boldsymbol{Q}^{(t)}_{\tilde{r}(t)} - \hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}(t)}\|_\infty\big] && \text{(C.57)}\\
&\quad + \mathbb{E}_{(t+1)}\big[\|\boldsymbol{Q}^{*}_{\tilde{r}(t+1)} - \boldsymbol{Q}^{*}_{\tilde{r}(t)}\|_\infty\big]\\
&\overset{(i)}{=} (1-\alpha)x_1^{(t)} + \alpha x_2^{(t)} + (1-\alpha)\mathbb{E}_{(t+1)}\big[\|\boldsymbol{Q}^{(t)}_{\tilde{r}(t)} - \hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}(t)}\|_\infty\big] && \text{(C.58)}\\
&\quad + \mathbb{E}_{(t+1)}\big[\|\boldsymbol{Q}^{*}_{\tilde{r}(t+1)} - \boldsymbol{Q}^{*}_{\tilde{r}(t)}\|_\infty\big].
\end{aligned}$$

In (i) we observe that the expectation *over the gradient step* of quantities that are independent of the gradient step is those quantities themselves. We are left with two separate expected "perturbation" terms which we will need to bound. First, recall that $\mathbb{E}_{(t+1)}\big[\|\boldsymbol{Q}^{(t)}_{\tilde{r}(t)} - \hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}(t)}\|_\infty\big]$ is a quantity we already know is bounded, indeed under Assumption 5.2.1, Lemma 5.2.2 states:

$$\mathbb{E}_{(t+1)}\big[\|\boldsymbol{Q}^{(t)}_{\tilde{r}(t)} - \hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}(t)}\|_\infty \leq \breve{\delta}. \tag{C.59}$$

Now onto the quantity $\mathbb{E}_{(t+1)}\big[\|\boldsymbol{Q}^{*}_{\tilde{r}(t+1)} - \boldsymbol{Q}^{*}_{\tilde{r}(t)}\|_\infty\big]$, we have already discussed (in Chapter 4) that it is Lipschitz with respect to the descent step in the variables $\boldsymbol{w}$ and $\boldsymbol{\lambda}$ (Proposition 4.3.4), we thus have:

$$\begin{aligned}
\mathbb{E}_{(t+1)}\big[\|\boldsymbol{Q}^{*}_{\tilde{r}(t+1)} - \boldsymbol{Q}^{*}_{\tilde{r}(t)}\|_\infty\big] &\overset{(i)}{\leq} 2(\|\Phi\| + \|\Psi\|)\mathbb{E}_{(t+1)}\big[\|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^{(t)}\|_2\big] && \text{(C.60)}\\
&= \eta_z C_z \mathbb{E}_{(t+1)}\big[\|\boldsymbol{g}_z^{(t)}\|_2\big] && \text{(C.61)}\\
&\overset{(ii)}{\leq} \eta_z C_z \Big(\mathbb{E}_{(t+1)}\big[\|\nabla_z L(\boldsymbol{\theta}^{(t)}, \boldsymbol{z}^{(t)})\|_\infty + \|\boldsymbol{g}_z^{(t)} - \nabla_z L(\boldsymbol{\theta}^{(t)}, \boldsymbol{z}^{(t)})\|_\infty\big]\Big) && \text{(C.62)}\\
&\overset{(iii)}{\leq} \eta_z C_z(\|\Phi\| + \|\Psi\| + \|\boldsymbol{b}\|_2 + \delta_z) = \eta_z(C'_z + C_z\delta_z). && \text{(C.63)}
\end{aligned}$$

Where in (i) we use Proposition 4.3.4, in (ii) we take a triangle inequality to isolate a deterministic exact gradient term from the error term. Finally, in (iii) we introduce Assumption 5.2.2 to bound the expected gradient. We then introduce $C'_z = 2(\|\Phi\| + \|\Psi\|)(\|\Phi\| + \|\Psi\| + \|\boldsymbol{b}\|_2)$. Plugging the bounds we just established back into (C.70) we get to the following result:

$$\mathbb{E}_{(t+1)}\big[x_2^{(t+1)}\big] \leq (1-\alpha)x_1^{(t)} + \alpha x_2^{(t)} + (1-\alpha)\breve{\delta} + \eta_z(C'_z + C_z\delta_z), \tag{C.64}$$

which completes the proof. □

Proof. We first consider Proposition 5.2.5, to do so, we start with the recursion defined in 5.38. Consider the vector $\boldsymbol{Q}^{*}_{\tilde{r}(t+1)} - \beta \log \boldsymbol{\xi}^{(t+1)}$:

$$\begin{aligned}
\boldsymbol{Q}^{*}_{\tilde{r}(t+1)} - \beta \log \boldsymbol{\xi}^{(t+1)} &\overset{(i)}{=} \boldsymbol{Q}^{*}_{\tilde{r}(t+1)} - \alpha\beta \log \boldsymbol{\xi}^{(t)} - (1-\alpha)\hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}(t)} && \text{(C.65)}\\
&\overset{(ii)}{=} \alpha\big(\boldsymbol{Q}^{*}_{\tilde{r}(t)} - \beta \log \boldsymbol{\xi}^{(t)}\big) + (1-\alpha)\big(\boldsymbol{Q}^{*}_{\tilde{r}(t)} - \boldsymbol{Q}^{(t)}_{\tilde{r}(t)}\big) && \text{(C.66)}\\
&\quad + (1-\alpha)\big(\boldsymbol{Q}^{(t)}_{\tilde{r}(t)} - \hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}(t)}\big) - \big(\boldsymbol{Q}^{*}_{\tilde{r}(t+1)} - \boldsymbol{Q}^{*}_{\tilde{r}(t)}\big).
\end{aligned}$$

Where in (i) we plug in the update rule for the auxiliary sequence as defined in (5.38), and in (ii) we reorganize the equation to isolate quantities that are relevant to our study[2]. Taking $\|\cdot\|_\infty$ norm and using the triangle inequality we get:

$$\begin{aligned}
x_2^{(t+1)} = \|\boldsymbol{Q}^{*}_{\tilde{r}(t+1)} - \beta \log \boldsymbol{\xi}^{(t+1)}\|_\infty &\leq \alpha\|\boldsymbol{Q}^{*}_{\tilde{r}(t)} - \beta \log \boldsymbol{\xi}^{(t)}\|_\infty + (1-\alpha)\|\boldsymbol{Q}^{*}_{\tilde{r}(t)} - \boldsymbol{Q}^{(t)}_{\tilde{r}(t)}\|_\infty && \text{(C.67)}\\
&\quad + (1-\alpha)\|\boldsymbol{Q}^{(t)}_{\tilde{r}(t)} - \hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}(t)}\|_\infty + \|\boldsymbol{Q}^{*}_{\tilde{r}(t+1)} - \boldsymbol{Q}^{*}_{\tilde{r}(t)}\|_\infty\\
&\overset{(i)}{\leq} \alpha x_2^{(t)} + (1-\alpha)x_1^{(t)} && \text{(C.68)}\\
&\quad + (1-\alpha)\|\boldsymbol{Q}^{(t)}_{\tilde{r}(t)} - \hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}(t)}\|_\infty + \|\boldsymbol{Q}^{*}_{\tilde{r}(t+1)} - \boldsymbol{Q}^{*}_{\tilde{r}(t)}\|_\infty.
\end{aligned}$$

[2] It is easily verified that line (C.66) simplifies back into (C.65)

In (i) we just plug in the scalar error terms from their definitions (5.40) and (5.41). We now take expectations (over the algorithm step) on both sides to get:

$$\mathbb{E}_{(t+1)}\big[x_2^{(t+1)}\big] \leq (1-\alpha)\mathbb{E}_{(t+1)}\big[x_1^{(t)}\big] + \alpha\mathbb{E}_{(t+1)}\big[x_2^{(t)}\big] + (1-\alpha)\mathbb{E}_{(t+1)}\big[\|\boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}} - \hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}^{(t)}}\|_\infty\big] \tag{C.69}$$
$$+ \mathbb{E}_{(t+1)}\big[\|\boldsymbol{Q}^{*}_{\tilde{r}^{(t+1)}} - \boldsymbol{Q}^{*}_{\tilde{r}^{(t)}}\|_\infty\big]$$
$$\overset{(i)}{=} (1-\alpha)x_1^{(t)} + \alpha x_2^{(t)} + (1-\alpha)\mathbb{E}_{(t+1)}\big[\|\boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}} - \hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}^{(t)}}\|_\infty\big] \tag{C.70}$$
$$+ \mathbb{E}_{(t+1)}\big[\|\boldsymbol{Q}^{*}_{\tilde{r}^{(t+1)}} - \boldsymbol{Q}^{*}_{\tilde{r}^{(t)}}\|_\infty\big].$$

In (i) we observe that the expectation *over the gradient step* of quantities that are independent of the gradient step is those quantities themedelves. We are left with two separate expected "perturbation" terms which we will need to bound. First recall that $\mathbb{E}_{(t+1)}\big[\|\boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}} - \hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}^{(t)}}\|_\infty\big]$ is a quantity we already know is bounded, indeed under Assumption 5.2.1, Lemma 5.2.2 states:

$$\mathbb{E}_{(t+1)}\big[\|\boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}} - \hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}^{(t)}}\|_\infty \leq \breve{\delta}. \tag{C.71}$$

Now onto the quantity $\mathbb{E}_{(t+1)}\big[\|\boldsymbol{Q}^{*}_{\tilde{r}^{(t+1)}} - \boldsymbol{Q}^{*}_{\tilde{r}^{(t)}}\|_\infty\big]$, we have already discussed (in Chapter 4) that it is Lipschitz with respect to the descent step in the variables $\boldsymbol{w}$ and $\boldsymbol{\lambda}$ (Proposition 4.3.4), we thus have:

$$\mathbb{E}_{(t+1)}\big[\|\boldsymbol{Q}^{*}_{\tilde{r}^{(t+1)}} - \boldsymbol{Q}^{*}_{\tilde{r}^{(t)}}\|_\infty\big] \overset{(i)}{\leq} 2(\|\Phi\| + \|\Psi\|)\mathbb{E}_{(t+1)}\big[\|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^{(t)}\|_2\big] \tag{C.72}$$
$$= \eta_z C_z \mathbb{E}_{(t+1)}\big[\|\boldsymbol{g}_z^{(t)}\|_2\big] \tag{C.73}$$
$$\overset{(ii)}{\leq} \eta_z C_z \Big(\mathbb{E}_{(t+1)}\big[\|\nabla_z L(\boldsymbol{\theta}^{(t)}, \boldsymbol{z}^{(t)})\|_\infty + \|\boldsymbol{g}_z^{(t)} - \nabla_z L(\boldsymbol{\theta}^{(t)}, \boldsymbol{z}^{(t)})\|_\infty\big]\Big) \tag{C.74}$$
$$\overset{(iii)}{\leq} \eta_z C_z(\|\Phi\| + \|\Psi\| + \|\boldsymbol{b}\|_2 + \delta_z) = \eta_z(C'_z + C_z\delta_z). \tag{C.75}$$

Where in (i) we use Proposition 4.3.4, in (ii) we take a triangle inequality to isolate a deterministic exact gradient term from the error term. Finally, in (iii) we introduce Assumption 5.2.2 to bound the expected gradient. We then introduce $C'_z = 2(\|\Phi\| + \|\Psi\|)(\|\Phi\| + \|\Psi\| + \|\boldsymbol{b}\|_2)$. Plugging the bounds we just established back into (C.70) we get to the following result:

$$\mathbb{E}_{(t+1)}\big[x_2^{(t+1)}\big] \leq (1-\alpha)x_1^{(t)} + \alpha x_2^{(t)} + (1-\alpha)\breve{\delta} + \eta_z(C'_z + C_z\delta_z), \tag{C.76}$$

which completes the proof. □

C.3.2. Proof of Proposition 5.2.6 (Second auxiliary sequence term)

We now move on to Proposition 5.2.6.

Proof. Our derivation starts by considering the vector $-\big(\boldsymbol{Q}^{*}_{\tilde{r}^{(t+1)}}(s,a) - \beta\log\boldsymbol{\xi}^{(t+1)}(s,a)\big)$, and as we did in the contraction bound of $x_2^{(t+1)}$ by plugging in the auxiliary sequence step (5.38):

$$-\big(\boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t+1)}} - \beta\log\boldsymbol{\xi}^{(t+1)}\big) = -\boldsymbol{Q}^{*}_{\tilde{r}^{(t+1)}} + \alpha\beta\log\boldsymbol{\xi}^{(t)} + (1-\alpha)\hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}^{(t+1)}} \tag{C.77}$$
$$\overset{(i)}{=} -\alpha(\boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}} - \beta\log\boldsymbol{\xi}^{(t)}) + (1-\alpha)(\hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}}) \tag{C.78}$$
$$+ (\boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t)}}) + (\boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t+1)}}).$$

Where (i) is just a rearranging step and matches the previous line exactly. Next, we take expectations on both sides:

$$\mathbb{E}_{(t+1)}\Big[-\big(\boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t+1)}} - \beta\log\boldsymbol{\xi}^{(t+1)}\big)\Big] = -\alpha\mathbb{E}_{(t+1)}\Big[\boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}} - \beta\log\boldsymbol{\xi}^{(t)}\Big] + (1-\alpha)\mathbb{E}_{(t+1)}\Big[\hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}}\Big] \tag{C.79}$$
$$+ \mathbb{E}_{(t+1)}\Big[\boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t)}}\Big] + \mathbb{E}_{(t+1)}\Big[\boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t+1)}}\Big]$$
$$= -\alpha(\boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}} - \beta\log\boldsymbol{\xi}^{(t)}) + (1-\alpha)\mathbb{E}_{(t+1)}\Big[\hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}}\Big] \tag{C.80}$$
$$+ \mathbb{E}_{(t+1)}\Big[\boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t)}}\Big] + \mathbb{E}_{(t+1)}\Big[\boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t+1)}}\Big].$$

Where in (i), we just drop expectations from the terms that are deterministic over the randomness of the algorithm step. Picking out the maximal $(s,a) \in S \times A$ pair on both sides of the inequality, we reach the expression of $x_3^{(t+1)}$ on the left hand side:

$$\mathbb{E}_{(t+1)}[x_3^{(t+1)}] = \mathbb{E}_{(t+1)}\Big[- \min_{s,a} \big(Q^{(t+1)}_{\tilde{r}^{(t+1)}}(s,a) - \beta \log \xi^{(t+1)}(s,a)\big)\Big] \tag{C.81}$$

$$= \min_{s,a} \Bigg[- \alpha(Q^{(t)}_{\tilde{r}^{(t)}}(s,a) - \beta \log \xi^{(t)}(s,a)) + (1-\alpha)\mathbb{E}_{(t+1)}\Big[\hat{Q}^{(t)}_{\tilde{r}^{(t)}}(s,a) - Q^{(t)}_{\tilde{r}^{(t)}}(s,a)\Big] \tag{C.82}$$

$$+ \mathbb{E}_{(t+1)}\Big[Q^{(t)}_{\tilde{r}^{(t)}}(s,a) - Q^{(t+1)}_{\tilde{r}^{(t)}}(s,a)\Big] + \mathbb{E}_{(t+1)}\Big[Q^{(t+1)}_{\tilde{r}^{(t)}}(s,a) - Q^{(t+1)}_{\tilde{r}^{(t+1)}}(s,a)\Big]\Bigg]$$

$$\leq -\alpha \min_{s,a}(Q^{(t)}_{\tilde{r}^{(t)}}(s,a) - \beta \log \xi^{(t)}(s,a)) + (1-\alpha)\mathbb{E}_{(t+1)}\Big[\|\hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}}\|_\infty\Big] \tag{C.83}$$

$$+ \mathbb{E}_{(t+1)}\Big[\|\boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t)}}\|_\infty\Big] + \mathbb{E}_{(t+1)}\Big[\|\boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t+1)}}\|_\infty\Big]$$

We have four terms which we will all bound with results that we already have established:

$$-\alpha \min_{s,a}(Q^{(t)}_{\tilde{r}^{(t)}}(s,a) - \beta \log \xi^{(t)}(s,a)) = \alpha x_3^{(t)} \qquad \text{by Definition of } x_3^{(t)}, \tag{C.84}$$

$$\mathbb{E}_{(t+1)}\Big[\|\hat{\boldsymbol{Q}}^{(t)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}}\|_\infty\Big] \leq \breve{\delta} \qquad \text{by Lemma 5.2.2,} \tag{C.85}$$

$$\mathbb{E}_{(t+1)}\Big[\|\boldsymbol{Q}^{(t)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t)}}\|_\infty\Big] \leq 2\gamma\breve{\delta} \qquad \text{by Lemma 5.2.2 and Corollary 5.2.3.1,} \tag{C.86}$$

$$\mathbb{E}_{(t+1)}\Big[\|\boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t)}} - \boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t+1)}}\|_\infty\Big] \leq \eta_z(C'_z + C_z\delta_z) \qquad \text{by Proposition 4.3.3 and then as (C.75).} \tag{C.87}$$

Putting everything back together completes the proof:

$$\mathbb{E}_{(t+1)}[x_3^{(t+1)}] \leq \alpha x_3^{(t)} + (\breve{\delta}(1+2\gamma) + \eta_z(C'_z + C_z\delta_z). \tag{C.88}$$

□

C.3.3. Proof of Proposition 5.2.4 (Q-value term)

Proof. Here our approach will be different from what we did for the last two contraction-bounds, we first look at any $(s,a) \in S \times A$, and we have that:

$$Q^*_{\tilde{r}^{(t+1)}}(s,a) - Q^{(t+1)}_{\tilde{r}^{(t+1)}}(s,a) = \big((1-\gamma)\tilde{r}(s,a) + \gamma\mathbb{E}_{s'|s,a}[V^*_{\tilde{r}^{(t+1)}}(s')]\big) \tag{C.89}$$

$$- \big((1-\gamma)\tilde{r}(s,a) + \gamma\mathbb{E}_{s'|s,a}[V^{(t+1)}_{\tilde{r}^{(t+1)}}(s')]\big)$$

$$\overset{(i)}{=} \gamma\mathbb{E}_{s'|s,a}[V^*_{\tilde{r}^{(t+1)}}(s')] - \gamma\mathbb{E}_{s'|s,a}[V^{(t+1)}_{\tilde{r}^{(t+1)}}(s')] \tag{C.90}$$

$$\overset{(ii)}{=} \gamma\mathbb{E}_{s'|s,a}[\beta \log \| \exp(\boldsymbol{Q}^*_{\tilde{r}^{(t+1)}}(s',\cdot)/\beta)\|_1] \tag{C.91}$$

$$- \gamma\mathbb{E}_{s',a'|s,a,\pi^{(t+1)}}\big[Q^{(t+1)}_{\tilde{r}^{(t+1)}}(s,a') - \beta \log \pi^{(t+1)}(a'|s')\big]$$

$$= \gamma\mathbb{E}_{s'|s,a}\big[\beta \log \| \exp(\boldsymbol{Q}^*_{\tilde{r}^{(t+1)}}(s',\cdot)/\beta)\|_1 - \beta \log \|\boldsymbol{\xi}^{(t+1)}(s,\cdot)\|_1\big] \tag{C.92}$$

$$- \gamma\mathbb{E}_{s',a'|s,a,\pi^{(t+1)}}\big[Q^{(t+1)}_{\tilde{r}^{(t+1)}}(s,a') - \beta(\alpha \log \xi^{(t)}(s',a')$$

$$+ (1-\alpha)/\beta\hat{Q}^{(t)}_{\tilde{r}^{(t)}}(s',a'))\big].$$

In (i) we plug in the expression of the optimal V-value from the optimal Q-values (equation (2.62)) and the expression of the V-value from the Q-value of some policies $\boldsymbol{\pi}^{(t+1)}$ (as in (2.62)). In (ii), we use the expression of the optimal policy in entropy-regularized *MDP*s 2.2.4 and the expression of the V-value as a function of the Q-value (equation (2.61)). Which, together with Proposition A.1.1, gives a bound in terms of $\|\cdot\|_\infty$ norms:

$$x_1^{(t+1)} = \gamma\|\boldsymbol{Q}^*_{\tilde{r}^{(t+1)}} - \boldsymbol{Q}^{(t+1)}_{\tilde{r}^{(t+1)}}\|_\infty \tag{C.93}$$

$$\leq \gamma \|\boldsymbol{Q}^*_{\tilde{r}^{(t+1)}} - \beta \log \boldsymbol{\xi}^{(t+1)}\|_\infty - \gamma \min_{s,a} \left(Q^{(t+1)}_{\tilde{r}^{(t+1)}}(s,a') - \beta \log \xi^{(t+1)}(s',a') \right) \tag{C.94}$$

$$= \gamma \left(x_2^{(t+1)} + x_3^{(t+1)} \right). \tag{C.95}$$

Here we thus clearly identify the terms $x_2^{(t+1)}$ and $x_3^{(t+1)}$. This is why we only tackle the contraction bound after having bounded the contraction rate of those two terms, and also what motivates the definition of the auxiliary sequence. Taking expectations with respect to the algorithm step on both sides we get:

$$\mathbb{E}_{(t+1)}[x_1^{(t+1)}] \leq \gamma \left(\mathbb{E}_{(t+1)}[x_2^{(t+1)}] + \mathbb{E}_{(t+1)}[x_3^{(t+1)}] \right) \tag{C.96}$$

$$\overset{(i)}{\leq} \gamma((1-\alpha)x_1^{(t)} + \alpha x_2^{(t)} + \alpha x_3^{(t)} + (\breve{\delta}(2+2\gamma-\alpha) + 2\eta_z(C'_z + C_z\delta_z)), \tag{C.97}$$

where (i) comes from plugging the contraction bounds from propositions 5.2.5 and 5.2.6. The proof is complete. □

D

Omitted proofs and derivations from Chapters 6

This appendix chapter details the results, proofs and derivations omitted from chapter 6.

D.1. A discussion on dual smoothness and dual strong convexity

In this additional section to the main body of text, we provide a proof that - if the right assumptions are met - the dual D:

$$D(\boldsymbol{r}, \boldsymbol{\lambda}) = \sup_{\boldsymbol{\theta}} \left(J(\boldsymbol{\theta}, \boldsymbol{r}) - J(\boldsymbol{\theta}^E, \boldsymbol{r}) + \langle \boldsymbol{\lambda}, \boldsymbol{b} - \Psi^\top \boldsymbol{\mu}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}} \rangle \right), \tag{D.1}$$

of our Lagrangian L:

$$L(\boldsymbol{\theta}, \boldsymbol{r}, \boldsymbol{\lambda}) = J(\boldsymbol{\theta}, \boldsymbol{r}) - J(\boldsymbol{\theta}^E, \boldsymbol{r}) + \langle \boldsymbol{\lambda}, \boldsymbol{b} - \Psi^\top \boldsymbol{\mu}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}} \rangle, \tag{D.2}$$

is a strongly convex function. We then discuss conditions for smoothness of the dual.

D.1.1. Dual strong convexity

We introduce a proof of dual strong-convexity that builds upon conditions for strong convexity of the Legendre-Fenchel conjugate (*LFC*). Recall the definition of the Legendre-Fenchel conjugate.

Definition D.1.1 (Legendre-Fenchel conjugate)**.** *Consider a strongly convex function $\tilde{\Omega} : X \to \mathbb{R}$. We call the function $\tilde{\Omega}^* : A \to \mathbb{R}$ defined as, for $a \in A$:*

$$\tilde{\Omega}^*(\boldsymbol{a}) = \sup_{\boldsymbol{x} \in X} \left[\langle \boldsymbol{a}, \boldsymbol{x} \rangle - \tilde{\Omega}(\boldsymbol{x}) \right], \tag{D.3}$$

the Legendre-Fenchel conjugate of $\tilde{\Omega}$.

A key property that we will need concerns conditions for the strong convexity of the *LFC*.

Proposition D.1.1 (Strong convexity of the Legendre Fenchel Conjugate)**.** *Consider the strongly convex function $\tilde{\Omega} : X \to \mathbb{R}$, its Legendre-Fenchel conjugate (Definition D.1.1) is strongly convex with some strong-convexity constant $C_{SD} > 0 \iff$ the function $\tilde{\Omega} : X \to \mathbb{R}$ is:*

1. *differentiable,*
2. *smooth with constant $1/C_{SC}$.*

This is a classical convex optimization result, a proof can be found in [Rockafellar and Wets 1998], proposition 12.60.

We also introduce a proposition about the composition of strongly convex functions and linear operators.

Proposition D.1.2 (Strong convexity is preserved by full-rank linear operators)**.** *Consider a α-strongly convex function $f : Y \to \mathbb{R}$ and a linear map $A : X \to Y$. If the linear map satisfies ker$(A^\top A) = 0$ (is full-rank) $\Rightarrow$ the composition $g = f \circ A : X \to \mathbb{R}$ is strongly convex with constant $\alpha\lambda_{\min}(A^\top A)$.*

A proof can be found in proposition 2.5 of [Guigues 2020a]

We are now ready to introduce the assumptions required to establish strong dual convexity. We require that the regularizer of our MDP be smooth and that the linear operator defined by our reward feature matrix and by our cost matrix be full-rank.

Assumption D.1.1 (Smooth regularizer)**.** *Assume that $\tilde{\Omega}$ is $1/\upsilon$-smooth, i.e.*

$$\|\nabla\tilde{\Omega}(\boldsymbol{\mu})\|_2 \leq \frac{1}{\upsilon}, \tag{D.4}$$

for any $\boldsymbol{\mu} \in$ dom$(\boldsymbol{\mu})$.

Assumption D.1.2 (Full rank linear operator)**.** *Assume that the linear operator $\mathcal{A}$ which we define as*

$$\mathcal{A} = \begin{bmatrix} \Phi & -\Psi^\top \end{bmatrix}, \tag{D.5}$$

is full-rank (ker$(A^T A) = 0$).

We are now ready to show if our assumptions are satisfied, the dual is a strongly convex function.

Proposition D.1.3 (Strong convexity of the dual)**.** *Assuming assumptions D.1.1 and D.1.2 are satisfied, the dual function of the Lagrangian L defined in (3.15) is strongly-convex with constant $C_{SC} := \upsilon\lambda_{min}(\mathcal{A}^T\mathcal{A})$.*

Proof. This analysis follows reasoning similar to the one described in [Guigues 2020b]. Starting with the of the Lagrangian we have:

$$L(\boldsymbol{\mu}, \boldsymbol{w}, \boldsymbol{\lambda}) := \langle\boldsymbol{\lambda}, \boldsymbol{b}\rangle - \langle\Phi\boldsymbol{w}, \boldsymbol{\mu}^E\rangle + \langle\boldsymbol{\mu}, \Phi\boldsymbol{w} - \boldsymbol{\lambda}\Psi\rangle - \tilde{\Omega}(\boldsymbol{\mu}). \tag{D.6}$$

Taking the dual we isolate the expression of the Legendre Fenchel Conjugate:

$$D(\boldsymbol{w}, \boldsymbol{\lambda}) = \langle\boldsymbol{\lambda}, \boldsymbol{b}\rangle - \langle\Phi\boldsymbol{w}, \boldsymbol{\mu}^E\rangle + \sup_{\boldsymbol{\mu}\in\mathcal{M}} \Big[\langle\boldsymbol{\mu}, \Phi\boldsymbol{w} - \boldsymbol{\lambda}\Psi\rangle - \tilde{\Omega}(\boldsymbol{\mu})\Big] \tag{D.7}$$

$$= \langle\boldsymbol{\lambda}, \boldsymbol{b}\rangle - \langle\Phi\boldsymbol{w}, \boldsymbol{\mu}^E\rangle + \tilde{\Omega}^*\big(\Phi\boldsymbol{w} - \boldsymbol{\lambda}\Psi\big). \tag{D.8}$$

By Assumption 6.1.1 we know that $\tilde{\Omega}^* : \mathbb{R}^{nm} \to \mathbb{R}$ is υ-strongly convex $\iff$ $\tilde{\Omega}$ is smooth with constant $1/\upsilon$ (which we impose with Assumption D.1.1). So in order to ensure that our dual indeed is strongly convex we just need to check that the linear map:

$$\Phi\boldsymbol{w} - \boldsymbol{\lambda}\Psi = \begin{bmatrix} \Phi & -\Psi^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{\lambda}^T \end{bmatrix} = \mathcal{A} \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{\lambda}^T \end{bmatrix}$$

is full rank, i.e. that ker$(\mathcal{A}^T\mathcal{A}) = 0$. We ensure that this is the case by Assumption D.1.2. By Proposition D.1.2 we have that D is strongly convex with constant $\upsilon\lambda_{\min}(\mathcal{A}^T\mathcal{A})$. □

D.1.2. Dual smoothness

We state a result about dual smoothness, which is borrowed from [Ying, Y. Ding, and Lavaei 2022]. Assuming that no state in the MDP is unvisited (which is generally considered a reasonable assumption), then dual smoothness follows.

Proposition D.1.4 (Dual smoothness)**.** *Given that Assumption 4.2.4, the dual function D of the Lagrangian (P2) is differentiable and smooth with parameter $L_z = \frac{2\ln(2)(nd+(1-\gamma)^2\sqrt{nd})}{\beta(1-\gamma)^3 c_v}$.*

This result is proved in [Ying, Y. Ding, and Lavaei 2022] for RL in the CMDP setting, it is equivalently true in the CIRL.

D.2. Affine error system for fast convergence

In the following appendix, we work our way towards proving Lemma 6.2.1. To do so, we decompose the lemma into 3 distinct propositions that we then bring together into the main result.

Let us first define a few quantities relevant to our analysis, as in the analysis of Lemma 5.2.4 we let:

$$\alpha = 1 - \eta_\theta \beta \tag{D.9}$$

and define the auxiliary sequence $\{\boldsymbol{\xi}^{(t)}\}_{t=0}^{T-1}$, $\boldsymbol{\xi}^{(t)} \in \mathbb{R}^{nm}$ recursively as follows:

$$\xi^{(0)}(s,a) := \|\boldsymbol{Q}^*_{\tilde{r}^{(0)}}(s,\cdot)/\beta\|_1 \cdot \pi^0(a|s) \tag{D.10}$$

$$\xi^{(t+1)}(s,a) := \left(\xi^{(t)}(s,a)\right)^\alpha \exp\left(\frac{1-\alpha}{\beta} Q^{(t)}_{\tilde{r}^{(t)}}(s,a)\right). \tag{D.11}$$

Recall that by definition, this auxiliary sequence satisfies

$$\boldsymbol{\pi}^{(t)}(a|s) = \frac{\xi^{(t)}(s,a)}{\sum_{a' \in A} \xi^{(t)}(s,a')}. \tag{D.12}$$

We will consider three scalar error terms:

$$x_1^{(t)} = \|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2 \tag{D.13}$$

$$x_2^{(t)} = \|\boldsymbol{Q}^{(t)}_{\boldsymbol{r}^{\tilde{(t)}}} - \boldsymbol{Q}^*_{\boldsymbol{r}^{\tilde{(t)}}}\|_\infty \tag{D.14}$$

$$x_3^{(t)} = \|\boldsymbol{Q}^*_{\boldsymbol{r}^{\tilde{(t)}}} - \beta \log \boldsymbol{\xi}^{(t)}\|_\infty \tag{D.15}$$

We will prove three propositions, each describing the evolution of the error terms as for one step of Algorithm 2.

Proposition D.2.1. *Under assumptions 4.2.1 and 4.2.2, NPG steps yield the following error-reduction bound:*

$$x_1^{(t+1)} \leq \left(1 - C_{\mathsf{SC}}\eta_z\right) x_1^{(t)} + C_\sigma \eta_z x_3^{(t)} \tag{D.16}$$

Proposition D.2.2. *Under assumptions 4.2.1 and 4.2.2, NPG steps yield the following error-reduction bound:*

$$x_2^{(t+1)} \leq \eta_z \frac{\gamma C_z L_z}{2b} x_1^{(t)} + \gamma x_2^{(t)} + 2\alpha\gamma x_3^{(t)} + 2\gamma b \tag{D.17}$$

Proposition D.2.3. *Under assumptions 4.2.1, 4.2.2, 6.1.1 and 6.1.2, NPG steps yield the following error-reduction bound:*

$$x_3^{(t+1)} \leq \eta_z \frac{C_z L_z}{4b} x_1^{(t)} + (1-\alpha) x_2^{(t)} + \alpha x_3^{(t)} + b \tag{D.18}$$

We prove that the statements of Proposition D.2.1, D.2.2 and D.2.3 in the next three subsections to this one, but we first show how they lead to the result of Lemma 6.2.1.

Proof. Recall that Lemma 6.2.1 makes use of a constant C defined as:

$$C = \max\left\{\frac{\gamma C_z L_z}{2b}, \frac{C_z L_z}{4b}, C_\sigma\right\}. \tag{D.19}$$

This constant should be understood as an upper bound on the influence of perturbations on both gradient ascent and gradient descent. It is determined by factors related to the CMDP properties

and the reward and constraint matrices. Furthermore since, $0 < \eta_z \leq 1/C_{\text{SC}}$ and $0 < \eta_\theta \leq 1/\beta$ we will renormalize our learning rates as follows:

$$\tilde{\eta}_z = \frac{\eta_z}{C_{\text{SC}}} \in (0, 1] \qquad \tilde{\eta}_\theta = \frac{\eta_\theta}{\beta} \in (0, 1]. \tag{D.20}$$

Using C as an upper bound for all problem-related constants and using $\tilde{\eta}_z$ and $\tilde{\eta}_\theta$ instead of their unnormalized forms, we can write Propositions D.2.1, D.2.2 and D.2.3 in a matrix form:

$$\begin{bmatrix} x_1^{(t+1)} \\ x_2^{(t+1)} \\ x_3^{(t+1)} \end{bmatrix} \leq \begin{bmatrix} (1-\tilde{\eta}_z) & 0 & C\tilde{\eta}_z \\ C\tilde{\eta}_z & \gamma & 2\gamma(1-\tilde{\eta}_\theta) \\ C\tilde{\eta}_z & \tilde{\eta}_\theta & (1-\tilde{\eta}_\theta) \end{bmatrix} \begin{bmatrix} x_1^{(t)} \\ x_2^{(t)} \\ x_3^{(t)} \end{bmatrix} + \begin{bmatrix} 0 \\ 2\gamma b \\ b \end{bmatrix}. \tag{D.21}$$

Which completes the proof of lemma 6.2.1. □

D.2.1. Proof of proposition D.2.3 (Auxiliary sequence term)

Proof. We first prove Proposition D.2.3, to do so we consider, for any $(s, a) \in S \times A$:

$$\begin{aligned} Q^*_{\tilde{\boldsymbol{r}}^{(t+1)}}(s,a) - \beta \log \xi^{(t+1)}(s,a) &\overset{(i)}{=} Q^*_{\tilde{\boldsymbol{r}}^{(t+1)}}(s,a) - \alpha\beta \log \xi^{(t)}(s,a) - (1-\alpha) Q^{(t)}_{\tilde{\boldsymbol{r}}}(s,a) && \text{(D.22)} \\ &\overset{(ii)}{=} Q^*_{\tilde{\boldsymbol{r}}^{(t)}}(s,a) - \alpha\beta \log \xi^{(t)}(s,a) - (1-\alpha) Q^{(t)}_{\tilde{\boldsymbol{r}}}(s,a) && \text{(D.23)} \\ &\quad - Q^*_{\tilde{\boldsymbol{r}}^{(t)}}(s,a) + Q^*_{\tilde{\boldsymbol{r}}^{(t+1)}}(s,a) \\ &\overset{(ii)}{=} \alpha\big(Q^*_{\tilde{\boldsymbol{r}}^{(t)}}(s,a) - \beta \log \xi^{(t)}(s,a)\big) && \text{(D.24)} \\ &\quad - (1-\alpha)\big(Q^*_{\tilde{\boldsymbol{r}}^{(t)}}(s,a) - Q^{(t)}_{\tilde{\boldsymbol{r}}}(s,a)\big) \\ &\quad - \big(Q^*_{\tilde{\boldsymbol{r}}^{(t)}}(s,a) + Q^*_{\tilde{\boldsymbol{r}}^{(t+1)}}(s,a)\big). \end{aligned}$$

Where (i) is obtained by plugging the expression (D.11) of the auxiliary sequence step into $\xi^{(t+1)}(s,a)$ and (ii) is simply obtained by adding $Q^*_{\tilde{\boldsymbol{r}}^{(t)}}(s,a) - Q^*_{\tilde{\boldsymbol{r}}^{(t)}}(s,a)$ and rearranging. Taking infinity norms and applying the triangle inequality we have:

$$\begin{aligned} \|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t+1)}} - \beta \log \boldsymbol{\xi}^{(t+1)}\|_\infty &\leq \alpha \|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t)}} - \beta \log \boldsymbol{\xi}^{(t)}\|_\infty + (1-\alpha)\|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}}\|_\infty && \text{(D.25)} \\ &\quad + \|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t+1)}}\|_\infty \\ &\overset{(i)}{\leq} \alpha \|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t)}} - \beta \log \boldsymbol{\xi}^{(t)}\|_\infty + (1-\alpha)\|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}}\|_\infty && \text{(D.26)} \\ &\quad + \frac{C_z L_z \eta_z}{4b} \|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2^2 + b. \end{aligned}$$

Where in (i) we upper bound $\|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t+1)}}\|_\infty$ using Proposition 4.3.4 as follows:

$$\begin{aligned} \|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t+1)}}\|_\infty &\overset{(i)}{\leq} C_z \|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^{(t)}\|_2 = C_z \eta_z \|\nabla_z L(\boldsymbol{\theta}^{(t)}, \boldsymbol{z}^{(t)})\|_2 && \text{(D.27)} \\ &\overset{(ii)}{\leq} C_z L_z \eta_z \|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2 \overset{(iii)}{\leq} \frac{C_z L_z \eta_z}{4b} \|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2^2 + b, && \text{(D.28)} \end{aligned}$$

(i) is the statement from Proposition 4.3.4 and (ii) holds as a result of Lipschitzness of the Lagrangian in $\boldsymbol{z}$. Finally, in (iii) we use that $x \leq b + x^2/(4b)$ for any $b > 0$. Identifying the error terms $x_1^{(t)}$, $x_2^{(t)}$, $x_3^{(t)}$ and $x_2^{(t+1)}$ we in (D.26) we reach the desired inequality. □

D.2.2. Proof of proposition D.2.2 (Q-value term)

Proof. Consider, for any $(s, a) \in S \times A$:

$$\begin{aligned} Q^*_{\tilde{\boldsymbol{r}}^{(t+1)}}(s,a) - Q^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t+1)}}(s,a) &\overset{(i)}{=} (1-\gamma)\tilde{r}^{(t+1)}(s,a) + \gamma \mathbb{E}_{s'|s,a}\Big[V^*_{\tilde{\boldsymbol{r}}^{(t+1)}}(s')\Big] && \text{(D.29)} \\ &\quad - (1-\gamma)\tilde{r}^{(t+1)}(s,a) - \gamma \mathbb{E}_{\substack{s'|s,a \\ a' \sim \boldsymbol{\pi}(\cdot|s')}}\Big[V^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t+1)}}(s')\Big] \end{aligned}$$

$$\overset{(ii)}{=}\gamma\mathbb{E}_{s'|s,a}\Big[V^*_{\tilde{\boldsymbol{r}}^{(t+1)}}(s')\Big]-\gamma\mathbb{E}_{\substack{s'|s,a\\ a'\sim\boldsymbol{\pi}^{(t+1)}(\cdot|s')}}\Big[V^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t+1)}}(s')\Big] \tag{D.30}$$

$$=\gamma\mathbb{E}_{s'|s,a}\left[\beta\log\left\|\exp\left(\frac{\boldsymbol{Q}_{\tilde{\boldsymbol{r}}^{(t+1)}(s'|\cdot)}}{\beta}\right)\right\|_1\right] \tag{D.31}$$

$$-\gamma\mathbb{E}_{\substack{s'|s,a\\ a'\sim\boldsymbol{\pi}^{(t+1)}(\cdot|s')}}\Big[Q^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t+1)}}(s',a')-\beta\log\pi^{(t+1)}(a'|s')\Big].$$

Where we obtain (i) obtained from (2.61) and (ii) from using equation (2.62) on the term $V^*_{\tilde{\boldsymbol{r}}^{(t+1)}}(s,a)$ and (2.62) on the term $V^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t+1)}}(s,a)$. We now focus on:

$$Q^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t+1)}}(s,a)-\beta\log\pi^{(t+1)}(a|s), \tag{D.32}$$

using the relationship (D.12) between the policy and the auxiliary sequence it naturally expands into:

$$Q^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t+1)}}(s,a)-\beta\big(\log\xi^{(t+1)}(a|s)-\log\|\boldsymbol{\xi}^{(t+1)}(\cdot|s)\|_1\big). \tag{D.33}$$

Introducing the auxiliary sequence step formulation reach:

$$Q^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t+1)}}(s,a)-\alpha\beta\log\big(\xi^{(t)}(s,a)\big)-(1-\alpha)Q^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a)+\beta\log\|\boldsymbol{\xi}^{(t+1)}(\cdot|s)\|_1. \tag{D.34}$$

Inserting (D.34) into (D.31) and using that $\beta\log\|\xi^{(t)}(\cdot|s)\|_1$ is independent of the action choice we have:

$$\begin{aligned}
&Q^*_{\tilde{\boldsymbol{r}}^{(t+1)}}(s,a)-Q^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t+1)}}(s,a)\\
&\quad=\gamma\mathbb{E}_{s'|s,a}\left[\beta\log\left\|\exp\left(\frac{\boldsymbol{Q}_{\tilde{\boldsymbol{r}}^{(t+1)}(s'|\cdot)}}{\beta}\right)\right\|_1-\beta\log\|\xi^{(t+1)}(\cdot|s)\|_1\right] && \text{(D.35)}\\
&\qquad-\gamma\mathbb{E}_{\substack{s'|s,a\\ a'\sim\boldsymbol{\pi}^{(t+1)}(\cdot|s')}}\Big[Q^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t+1)}}(s',a')-\alpha\beta\log\big(\xi^{(t)}(s',a')\big)-(1-\alpha)Q^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s',a')\Big]\\
&\quad\overset{(i)}{\leq}\gamma\|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t+1)}}-\beta\boldsymbol{\xi}^{(t+1)}\|_\infty && \text{(D.36)}\\
&\qquad-\gamma\mathbb{E}_{\substack{s'|s,a\\ a'\sim\boldsymbol{\pi}^{(t+1)}(\cdot|s')}}\Big[Q^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t+1)}}(s',a')-\alpha\beta\log\big(\xi^{(t)}(s',a')\big)-(1-\alpha)Q^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s',a')\Big]\\
&\quad=\gamma\|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t+1)}}-\beta\boldsymbol{\xi}^{(t+1)}\|_\infty+\gamma\mathbb{E}_{\substack{s'|s,a\\ a'\sim\boldsymbol{\pi}^{(t+1)}(\cdot|s')}}\Big[Q^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t)}}(s',a')-Q^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t+1)}}(s',a')\Big] && \text{(D.37)}\\
&\qquad-\gamma\mathbb{E}_{\substack{s'|s,a\\ a'\sim\boldsymbol{\pi}^{(t+1)}(\cdot|s')}}\Big[Q^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t)}}(s',a')-\alpha\beta\log\big(\xi^{(t)}(s',a')\big)-(1-\alpha)Q^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s',a')\Big]\\
&\quad=\gamma\|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t+1)}}-\beta\boldsymbol{\xi}^{(t+1)}\|_\infty+\gamma\|\boldsymbol{Q}^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t)}}-\boldsymbol{Q}^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t+1)}}\|_\infty && \text{(D.38)}\\
&\qquad-\gamma\mathbb{E}_{\substack{s'|s,a\\ a'\sim\boldsymbol{\pi}^{(t+1)}(\cdot|s')}}\Big[Q^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t)}}(s',a')-\alpha\beta\log\big(\xi^{(t)}(s',a')\big)-(1-\alpha)Q^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s',a')\Big]
\end{aligned}$$

Where (i) holds from Proposition A.1.1. We now turn our attention to:

$$Q^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a)-\alpha\beta\log\big(\xi^{(t)}(s,a)\big)-(1-\alpha)Q^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a). \tag{D.39}$$

Using that the Q-values are monotonously improving (Proposition 2.4.2) we have that:

$$\begin{aligned}
&Q^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a)-\alpha\beta\log\big(\xi^{(t)}(s,a)\big)-(1-\alpha)Q^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a) && \text{(D.40)}\\
&\geq Q^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a)-\alpha\beta\log\big(\xi^{(t)}(s,a)\big)-(1-\alpha)Q^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a) && \text{(D.41)}\\
&=\alpha\big(Q^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a)-\beta\log\xi^{(t)}(s,a)\big) && \text{(D.42)}\\
&=\alpha\big(Q^*_{\tilde{\boldsymbol{r}}^{(t)}}(s,a)-\beta\log\xi^{(t)}(s,a)\big)+\alpha\big(Q^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}(s,a)-Q^*_{\tilde{\boldsymbol{r}}^{(t)}}(s,a)\big) && \text{(D.43)}\\
&\geq-\alpha\big\|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t)}}-\beta\log\boldsymbol{\xi}^{(t)}\big\|_\infty-\alpha\big\|\boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}}-\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t)}}\big\|_\infty. && \text{(D.44)}
\end{aligned}$$

Inserting (D.44) into (D.39) we reach the form:

$$\begin{aligned}\|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t+1)}} - \boldsymbol{Q}^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t+1)}}\|_\infty \leq& \gamma\|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t+1)}} - \beta\boldsymbol{\xi}^{(t+1)}\|_\infty + \gamma\|\boldsymbol{Q}^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t+1)}}\|_\infty \qquad \text{(D.45)}\\ &+ \gamma\big(\alpha\big\|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t)}} - \beta\log\boldsymbol{\xi}^{(t)}\big\|_\infty + \alpha\big\|\boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t)}}\big\|_\infty\big)\\ \overset{(i)}{\leq}& \gamma\|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t+1)}} - \beta\boldsymbol{\xi}^{(t+1)}\|_\infty + \gamma\frac{C_z L_z \eta_z}{4b}\|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2 + \gamma b \qquad \text{(D.46)}\\ &+ \gamma\big(\alpha\big\|\boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t)}} - \beta\log\boldsymbol{\xi}^{(t)}\big\|_\infty + \alpha\big\|\boldsymbol{Q}^{(t)}_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^*_{\tilde{\boldsymbol{r}}^{(t)}}\big\|_\infty\big).\end{aligned}$$

We obtain the inequality (i) as follows:

$$\begin{aligned}\|\boldsymbol{Q}^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t)}} - \boldsymbol{Q}^{(t+1)}_{\tilde{\boldsymbol{r}}^{(t+1)}}\|_\infty \overset{(i)}{\leq}& C_z\|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^{(t)}\|_2 = C_z\eta_z\|\nabla_z L(\boldsymbol{\theta}^{(t)}, \boldsymbol{z}^{(t)})\|_2 \qquad \text{(D.47)}\\ \overset{(ii)}{\leq} C_z L_z \eta_z\|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2 \overset{(iii)}{\leq}& \frac{C_z L_z \eta_z}{4b}\|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2 + b, \qquad \text{(D.48)}\end{aligned}$$

where (i) holds by Proposition 4.3.3 and (ii) by Lipschitzness of the Lagrangian with respect to $\boldsymbol{z}$. We complete the verification of the inequality by using in (iii) that for any $b > 0$, $x < b + x^2/(4b)$. Equation (D.46) can now be expressed solely as a function of the error terms, which makes it more readable:

$$x_2^{(t+1)} \leq \gamma\eta_z C_z L_z x_1^{(t)} + \alpha\gamma x_2^{(t)} + \alpha\gamma x_3^{(t)} + \gamma x_3^{(t+1)} \overset{(i)}{\leq} \gamma\frac{C_z L_z \eta_z}{2b}x_1^{(t)} + \gamma x_2^{(t)} + 2\alpha\gamma x_3^{(t)} + 2\gamma b, \qquad \text{(D.49)}$$

where (i) holds by upper bounding the term $x_3^{(t+1)}$ with Proposition D.2.3. The inequality is verified. □

D.2.3. Proof of proposition D.2.1 (Dual term)

Finally, we consider proposition D.2.1. Before diving into the analysis, we state two useful propositions about smooth convex functions and about MDPs that we will require for our derivations.

Proposition D.2.4 (Sufficient decrease)**.** *Consider a function $D : \mathbb{R}^{k+d} \to \mathbb{R}$ be convex, differentiable and L_z-smooth, projected gradient descent steps with learning rate $\eta_z \leq \frac{1}{L_z}$ satisfies:*

$$D(\boldsymbol{z}^*) - D(\boldsymbol{z}^{(t)}) \leq D(\boldsymbol{z}^{(t+1)}) - D(\boldsymbol{z}^{(t)}) \leq -\frac{1}{2L}\|\nabla_x D(\boldsymbol{z}^{(t)})\|_2^2 + \frac{L_z}{2}\|\boldsymbol{z}^{(t+1/2)} - \boldsymbol{z}^{(t+1)}\|_2^2. \qquad \text{(D.50)}$$

The proof is a very common result from convex optimization we include it for completeness in Appendix D.4.

Proposition D.2.5 (Occupancy measure is Lipschitz with respect to the policies)**.** *The occupancy measures $\boldsymbol{\mu}_{\boldsymbol{\pi}}$ and $\boldsymbol{\mu}_{\bar{\boldsymbol{\pi}}}$ respectively induced by policies $\boldsymbol{\pi}$ and $\bar{\boldsymbol{\pi}}$ on MDPs with identical state space, action space and Markovian transition kernel satisfy the following inequality:*

$$\|\boldsymbol{\mu}_{\boldsymbol{\pi}} - \boldsymbol{\mu}_{\bar{\boldsymbol{\pi}}}\|_2 \leq B_\mu\|\boldsymbol{\pi} - \bar{\boldsymbol{\pi}}\|_2,$$

where $B_\mu = \frac{1}{1-\gamma}$. Proof in Appendix D.3.

We are now ready to dive into the proof of Proposition D.2.1.

Proof. First recall that we do not have access to the gradients of the dual function so using dual smoothness will not be trivial. We consider the decomposition of the gradients into the gradient of the dual $\nabla_z D(\boldsymbol{z}^{(t)})$ and a perturbation term $\boldsymbol{\sigma}_z^{(t)}$:

$$\nabla_z L(\boldsymbol{\theta}^{(t)}, \boldsymbol{z}^{(t)}) = \boldsymbol{g}_z^{(t)} = \nabla_z D(\boldsymbol{z}^{(t)}) + \boldsymbol{\sigma}_z^{(t)}. \qquad \text{(D.51)}$$

Now start our analysis from strong convexity of the dual function (Proposition D.1.1):

$$D(\boldsymbol{z}^{(t)}) - D^* + \frac{C_{\mathsf{SC}}}{2}\|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2^2 \overset{(i)}{\leq} \langle\nabla_z D(\boldsymbol{z}^{(t)}), \boldsymbol{z}^{(t)} - \boldsymbol{z}^*\rangle \qquad \text{(D.52)}$$

$$\overset{(ii)}{\leq} \langle \boldsymbol{g}_z^{(t)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle - \langle \boldsymbol{\sigma}_z^{(t)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle \tag{D.53}$$

$$= \frac{1}{2\eta_z} \Big(\|\boldsymbol{g}_z^{(t)}\|_2^2 + \|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2^2 - \|\boldsymbol{z}^{(t+1/2)} - \boldsymbol{z}^*\|_2^2 \Big) - \langle \boldsymbol{\sigma}_z^{(t)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle \tag{D.54}$$

$$\overset{(iii)}{\leq} \frac{1}{2\eta_z} \Big(\eta_z^2 \|\boldsymbol{g}_z^{(t)}\|_2^2 + \|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2^2 - \|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^*\|_2^2 - \|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^{(t+1/2)}\|_2^2 \Big) - \langle \boldsymbol{\sigma}_z^{(t)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle. \tag{D.55}$$

Where (i) holds by the strong convexity of the dual, (ii) is obtained from the gradient decomposition (D.51) and (iii) is true because projection is non-expansive. Rearranging (D.55) yields:

$$\begin{aligned} \|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^*\|_2^2 \leq \eta_z^2 \|\boldsymbol{g}_z^{(t)}\|_2^2 + 2\eta_z \big(D^* - D(\boldsymbol{z}^{(t)}) \big) - \|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^{(t+1/2)}\|_2^2 \\ + (1 - \eta_z C_{\mathsf{SC}}) \|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2^2 - 2\eta_z \langle \boldsymbol{\sigma}_z^{(t)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle. \end{aligned} \tag{D.56}$$

Paying specific attention to the term $\eta_z^2 \|\boldsymbol{g}_z^{(t)}\|_2^2 + 2\eta_z \big(D^* - D(\boldsymbol{z}^{(t)}) \big) - \|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^{(t+1/2)}\|_2^2$ we reach:

$$\eta_z^2 \|\boldsymbol{g}_z^{(t)}\|_2^2 + 2\eta_z \big(D^* - D(\boldsymbol{z}^{(t)}) \big) - \|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^{(t+1/2)}\|_2^2 \tag{D.57}$$

$$\overset{(i)}{=} \eta_z^2 \|\nabla_z D(\boldsymbol{z}^{(t)}) + \boldsymbol{\sigma}_z^{(t)}\|_2^2 - \|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^{(t+1/2)}\|_2^2 + 2\eta_z \big(D^* - D(\boldsymbol{z}^{(t)}) \big) \tag{D.58}$$

$$\begin{aligned} = \eta_z^2 \|\nabla_z D(\boldsymbol{z}^{(t)})\|_2^2 + 2\eta_z \big(D^* - D(\boldsymbol{z}^{(t)}) \big) - \|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^{(t+1/2)}\|_2^2 \\ + \eta_z^2 \|\boldsymbol{\sigma}_z^{(t)}\|_2^2 + 2\eta_z^2 \langle \nabla_z D(\boldsymbol{z}^{(t)}), \boldsymbol{\sigma}_z^{(t)} \rangle \end{aligned} \tag{D.59}$$

$$\overset{(ii)}{\geq} \eta_z^2 \|\boldsymbol{\sigma}_z^{(t)}\|_2^2 + 2\eta_z^2 \langle \nabla_z D(\boldsymbol{z}^{(t)}), \boldsymbol{\sigma}_z^{(t)} \rangle. \tag{D.60}$$

Where (i) holds by the gradient decomposition (D.51) and (ii) holds by sufficient decrease (Proposition D.2.4), specifically it can be verified by:

$$\eta_z^2 \|\nabla_z D(\boldsymbol{z}^{(t)})\|_2^2 + 2\eta_z \big(D^* - D(\boldsymbol{z}^{(t)}) \big) - \|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^{(t+1/2)}\|_2^2 \tag{D.61}$$

$$\overset{(i)}{\leq} \frac{1}{L_z^2} \|\nabla_z D(\boldsymbol{z}^{(t)})\|_2^2 + \frac{2}{L_z} \big(D^* - D(\boldsymbol{z}^{(t)}) \big) - \|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^{(t+1/2)}\|_2^2 \tag{D.62}$$

$$\overset{(ii)}{\leq} \frac{1}{L_z^2} \|\nabla_z D(\boldsymbol{z}^{(t)})\|_2^2 + \frac{2}{L_z} \left(- \frac{\|\nabla_z D(\boldsymbol{z}^{(t)})\|_2^2}{2L_z} + \frac{L_z}{2} \|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^{(t+1/2)}\|_2^2 \right) - \|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^{(t+1/2)}\|_2^2 \tag{D.63}$$

$$= 0. \tag{D.64}$$

Here (i) is obtained by plugging learning rate upper-bound $\eta_z \leq \frac{1}{L_z}$ and (ii) is sufficient decrease (Proposition D.2.4). Inserting (D.60) into (D.56) we reach:

$$\|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^*\|_2^2 \leq (1 - \eta_z C_{\mathsf{SC}}) \|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2^2 - 2\eta_z \langle \boldsymbol{\sigma}_z^{(t)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle + \eta_z^2 \|\boldsymbol{\sigma}_z^{(t)}\|_2^2 + 2\eta_z^2 \langle \nabla_z D(\boldsymbol{z}^{(t)}), \boldsymbol{\sigma}_z^{(t)} \rangle \tag{D.65}$$

We will separately consider terms (a), (b) and (c), starting with the term (a):

$$|\langle \boldsymbol{\sigma}_z^{(t)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^* \rangle| \overset{(i)}{\leq} \frac{C_z \sqrt{k+d} (1 + \gamma \sqrt{nm}) D_{1,z}}{2(1-\gamma)} \big\| \log \boldsymbol{\pi}_{\boldsymbol{\theta}^{(t)}} - \log \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}} \big\|_\infty \tag{D.66}$$

$$\overset{(ii)}{\leq} \frac{C_z \sqrt{k+d} (1 + \gamma \sqrt{nm}) D_{1,z}}{\beta(1-\gamma)} \| \log \boldsymbol{Q}_{\tilde{\boldsymbol{r}}^{(t)}} - \beta \log \boldsymbol{\xi}^{(t)} \|_\infty. \tag{D.67}$$

Where (i) holds by the exact same reasoning as (4.78), and (ii) holds because $\| \log \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}^{(t)}} - \log \boldsymbol{\pi}^{(t)} \|_\infty \leq 2 \| \log \boldsymbol{Q}_{\tilde{\boldsymbol{r}}^{(t)}} / \beta - \log \boldsymbol{\xi}^{(t)} \|_\infty = \frac{2}{\beta} \| \log \boldsymbol{Q}_{\tilde{\boldsymbol{r}}^{(t)}} - \beta \log \boldsymbol{\xi}^{(t)} \|_\infty$. Now moving on to term (b) we consider:

$$\|\boldsymbol{\sigma}_z^{(t)}\|_2^2 \overset{(i)}{\leq} \left(\frac{C_z}{2} \|\boldsymbol{\mu}^{(t)} - \boldsymbol{\mu}^*_{\tilde{\boldsymbol{r}}^{(t)}}\|_2 \right)^2 \tag{D.68}$$

$$\overset{(ii)}{\leq} \left(\frac{C_z}{2(1-\gamma)} \|\boldsymbol{\pi}^{(t)} - \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}^{(t)}}\|_2 \right)^2 \tag{D.69}$$

$$\overset{(iii)}{\leq} \left(\frac{\sqrt{nm} C_z}{2(1-\gamma)} \|\boldsymbol{\pi}^{(t)} - \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}^{(t)}}\|_\infty \right)^2 \tag{D.70}$$

$$\overset{(iv)}{\leq} \frac{C_z^2(nm)}{4(1-\gamma)^2} \| \log \boldsymbol{\pi}^{(t)} - \log \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}^{(t)}} \|^2_\infty \tag{D.71}$$

$$\overset{(v)}{\leq} \frac{C_z^2(nm)}{(1-\gamma)^2} \| \log \boldsymbol{\pi}^{(t)} - \log \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}^{(t)}} \|_\infty \tag{D.72}$$

$$\leq \frac{2C_z^2(nm)}{\beta(1-\gamma)^2} \|\boldsymbol{Q}_{\tilde{\boldsymbol{r}}^{(t)}} - \beta \log \boldsymbol{\xi}^{(t)}\|_\infty. \tag{D.73}$$

Where (i) holds by relating the perturbation term to the difference of occupancy measures by its definition (4.68) $\boldsymbol{\sigma}_z^{(t)} = [\Psi, \Phi]^T(\boldsymbol{\mu}^{(t)} - \boldsymbol{\mu}^*_{\tilde{\boldsymbol{r}}^{(t)}})$ and then upper bounding that by the sum of spectral norms $C_z = 2(\|\Psi\| + \|\Phi\|)$. Step (ii) leverages the Lipschitzness of the occupancy measure with respect to the policy (Proposition D.2.5). Then (iii) holds by $\|\cdot\|_2 \leq \sqrt{nm}\|\cdot\|_\infty$, (iv) is obtained by observing that the difference of logs upper bounds the difference of values on the relevant domain. Finally, (iv) uses that the $(x)^2$ function is Lipschitz with constants $2x_{\text{max}}$ since the maximum value and that the difference between two logs on that domain is upper bounded by 2. The last step holds by $\| \log \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}^{(t)}} - \log \boldsymbol{\pi}^{(t)}\|_\infty \leq 2\| \log \boldsymbol{Q}_{\tilde{\boldsymbol{r}}^{(t)}}/\beta - \log \boldsymbol{\xi}^{(t)}\|_\infty = \frac{2}{\beta}\| \log \boldsymbol{Q}_{\tilde{\boldsymbol{r}}^{(t)}} - \beta \log \boldsymbol{\xi}^{(t)}\|_\infty$.

We are now ready to consider (c), the last term. The derivation is very similar to (a):

$$|\langle \boldsymbol{\sigma}_z^{(t)}, \Delta_z D(\boldsymbol{z}^{(t)})\rangle| \overset{(i)}{\leq} \frac{C_z\sqrt{k+d}(1+\gamma\sqrt{nm})}{2(1-\gamma)} \left\| \log \boldsymbol{\pi}_{\boldsymbol{\theta}^{(t)}} - \log \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}} \right\|_\infty \|\nabla_z D(\boldsymbol{z}^{(t)})\|_1 \tag{D.74}$$

$$\overset{(ii)}{\leq} \frac{C_z\sqrt{k+d}(1+\gamma\sqrt{nm})\sqrt{nm}B_z}{\beta(1-\gamma)} \| \log \boldsymbol{Q}_{\tilde{\boldsymbol{r}}^{(t)}} - \beta \log \boldsymbol{\xi}^{(t)}\|_\infty. \tag{D.75}$$

Where (i) holds by the same reasoning as (4.78), (ii) holds because

$$\| \log \boldsymbol{\pi}^*_{\tilde{\boldsymbol{r}}^{(t)}} - \log \boldsymbol{\pi}^{(t)}\|_\infty \leq \frac{2}{\beta}\| \log \boldsymbol{Q}_{\tilde{\boldsymbol{r}}^{(t)}} - \beta \log \boldsymbol{\xi}^{(t)}\|_\infty, \tag{D.76}$$

and because the dual is Lipschitz with constants B_z. Bringing (a), (b) and (c) together we reach:

$$2\eta_z \overbrace{\langle \boldsymbol{\sigma}_z^{(t)}, \boldsymbol{z}^{(t)} - \boldsymbol{z}^*\rangle}^{(a)} + \eta_z^2 \overbrace{\|\boldsymbol{\sigma}_z^{(t)}\|_2^2}^{(b)} + 2\eta_z^2 \overbrace{\langle \nabla_z D(\boldsymbol{z}^{(t)}), \boldsymbol{\sigma}_z^{(t)}\rangle}^{(c)} \tag{D.77}$$

$$\overset{(i)}{\leq} \eta_z \left(\frac{4C_z^2(nm)}{\beta L_z (1-\gamma)^2} + \left(2D_{1,z} + \frac{4\sqrt{nm}B_z}{L_z}\right) \frac{C_z\sqrt{k+d}(1+\gamma\sqrt{nm})}{\beta(1-\gamma)} \right) \| \log \boldsymbol{Q}_{\tilde{\boldsymbol{r}}^{(t)}} - \beta \log \boldsymbol{\xi}^{(t)}\|_\infty \tag{D.78}$$

$$= C_\sigma x_3^{(t)}. \tag{D.79}$$

Where (i) holds because since $\eta_z \leq \frac{1}{L_z}$ we have $\eta_z^2 \leq \frac{2\eta_z}{L_z}$. Inserting (D.79) into (D.65) we reach the inequality of proposition D.2.1 and our proof is complete.

$$\|\boldsymbol{z}^{(t+1)} - \boldsymbol{z}^*\|_2^2 \leq \Big(1 - C_{\text{SC}}\eta_z\Big)\|\boldsymbol{z}^{(t)} - \boldsymbol{z}^*\|_2^2 + C_\sigma \eta_z x_3^{(t)} \tag{D.80}$$

$$= \Big(1 - C_{\text{SC}}\eta_z\Big) x_1^{(t)} + C_\sigma \eta_z x_3^{(t)}. \tag{D.81}$$

□

D.3. Proof of proposition D.2.5 (Occupancy measure is Lipschitz with respect to the policies)

We start by expanding the expression from the lefthand side of the inequality we are trying to prove:

$$\|\boldsymbol{\mu}_\pi - \boldsymbol{\mu}_{\bar{\pi}}\|_2 = \sqrt{\sum_{s,a \in S \times A} \Big(\mu_{\boldsymbol{\pi}}(s,a) - \mu_{\bar{\boldsymbol{\pi}}}(s,a)\Big)^2}$$

$$
\begin{aligned}
&\overset{(i)}{=} \sqrt{\sum_{s,a \in S \times A} \Big(\mu_{\boldsymbol{\pi}}(s)\pi(a|s) - \mu_{\bar{\boldsymbol{\pi}}}(s)\bar{\pi}(a|s)\Big)^2} \\
&\overset{(ii)}{=} \sqrt{\sum_{s,a \in S \times A} \Big(\mu_{\boldsymbol{\pi}}(s)\big(\pi(a|s) - \bar{\pi}(a|s)\big) + \bar{\pi}(a|s)\big(\mu_{\boldsymbol{\pi}}(s) - \mu_{\bar{\boldsymbol{\pi}}}(s)\big)\Big)^2} \\
&\overset{(iii)}{\leq} \overbrace{\sqrt{\sum_{s,a \in S \times A} \Big(\mu_{\boldsymbol{\pi}}(s)\big(\pi(a|s) - \bar{\pi}(a|s)\big)\Big)^2}}^{\leq \|\boldsymbol{\pi} - \bar{\boldsymbol{\pi}}\|_2} \\
&+ \overbrace{\sqrt{\sum_{s,a \in S \times A} \Big(\bar{\pi}(a|s)\big(\mu_{\boldsymbol{\pi}}(s) - \mu_{\bar{\boldsymbol{\pi}}}(s)\big)\Big)^2}}^{\leq \|\boldsymbol{\mu}_s - \bar{\boldsymbol{\mu}}_s\|_2} \\
&\overset{(iv)}{\leq} \|\boldsymbol{\pi} - \bar{\boldsymbol{\pi}}\|_2 + \|\boldsymbol{\mu}_s - \bar{\boldsymbol{\mu}}_s\|_2.
\end{aligned}
$$

Where in (i) we just plug in the defintion of the state-occupancy measure (def 2.1.9), in (ii) we just add $0 = \mu_{\boldsymbol{\pi}}(s)\bar{\pi}(a|s) - \mu_{\boldsymbol{\pi}}(s)\bar{\pi}(a|s)$ and rearrang. Next, we just use a triangle inequality (iii) and observing that both sides are upper bounded by l2 norms we are done with the first step.

We will now be concerned with bounding $\|\boldsymbol{\mu}_s - \bar{\boldsymbol{\mu}}_s\|_2$ by $\|\boldsymbol{\pi} - \bar{\boldsymbol{\pi}}\|_2$ (mutliplied by some constant term).

To do so we first show a useful result on the spectral norm of the difference between the inverse of two matrices:

$$
\begin{aligned}
\|A^{-1} + B^{-1}\| &\overset{(i)}{=} \|A^{-1}(A+B)B^{-1}\| \\
&\overset{(ii)}{\leq} \|A^{-1}\| \cdot \|(A+B)\| \cdot \|B^{-1}\| \\
&\overset{(iii)}{=} \frac{\|(A+B)\|}{\sigma_{\mathsf{min}}(A) \cdot \sigma_{\mathsf{min}}(B)}.
\end{aligned} \tag{l}
$$

Where (i) holds by the equality $A^{-1} + B^{-1} = A^{-1}(A+B)B^{-1}$ which holds for any two invertible matrices (a proof can be found in the solution handbook to Searle 1982), (ii) holds by submultiplicativity of the spectral norm and (iii) uses the definition of the spectal norm ($\sigma_{\mathsf{min}}(A)$ denotes the minimum eigenvalue of the matrix A).

We now get back to bounding $\|\boldsymbol{\mu}_s - \bar{\boldsymbol{\mu}}_s\|_2$:

$$
\begin{aligned}
\|\boldsymbol{\mu}_s - \bar{\boldsymbol{\mu}}_s\|_2 &\overset{(i)}{=} (1-\gamma)\left\|\Big[\big(I - \gamma P^{\boldsymbol{\pi}}\big)^{-1} - \big(I - \gamma P^{\bar{\boldsymbol{\pi}}}\big)^{-1}\Big]\boldsymbol{\nu}\right\|_2 \\
&\overset{(ii)}{\leq} (1-\gamma)\left\|\Big[\big(I - \gamma P^{\boldsymbol{\pi}}\big)^{-1} - \big(I - \gamma P^{\bar{\boldsymbol{\pi}}}\big)^{-1}\Big]\right\| \cdot \|\boldsymbol{\nu}\|_2 \\
&\overset{(iii)}{\leq} (1-\gamma)\left\|\Big[\big(I - \gamma P^{\boldsymbol{\pi}}\big)^{-1} - \big(I - \gamma P^{\bar{\boldsymbol{\pi}}}\big)^{-1}\Big]\right\| \\
&\overset{(iv)}{\leq} (1-\gamma)^{-1}\left\|\Big[\big(I - \gamma P^{\boldsymbol{\pi}}\big) - \big(I - \gamma P^{\bar{\boldsymbol{\pi}}}\big)\Big]\right\| \\
&= \frac{\gamma}{1-\gamma}\big\|P^{\boldsymbol{\pi}} - P^{\bar{\boldsymbol{\pi}}}\big\|.
\end{aligned}
$$

Where in (i) we just use the closed form computation of the state-occupancy measure from the policy (as shown in equation (2.32)), in (ii) we just use the definition of the spectral norm with pulls out the $\|\boldsymbol{\nu}\|_2$ term, which we know is smaller or equal to 1 (since it is the l2 norm of a probability distribution) which gives us inequality (iii). Now plugging in the result from (l) and using that the smallest

eigenvalue of $\left(I-\gamma P^{\boldsymbol{\pi}}\right)^{-1}$ is greater or equal to $1-\gamma$ we get inequality (iv) which simplifies into the last line.

Now we just need to bound $\left\|P^{\boldsymbol{\pi}}-P^{\bar{\boldsymbol{\pi}}}\right\|$ which we do as follows:

$$\begin{aligned}
\left\|P^{\boldsymbol{\pi}}-P^{\bar{\boldsymbol{\pi}}}\right\| &\overset{(i)}{\leq} \left\|P^{\boldsymbol{\pi}}-P^{\bar{\boldsymbol{\pi}}}\right\|_F \\
&\overset{(ii)}{=} \sqrt{\sum_{s,s'\in S\times S}\left(P^{\boldsymbol{\pi}}(s'|s)-P^{\bar{\boldsymbol{\pi}}}(s'|s)\right)^2} \\
&\overset{(iii)}{=} \sqrt{\sum_{s,s'\in S\times S}\Big(\sum_{a\in A}P(s'|s,a)(\pi(a|s)-\bar{\pi}(a|s))\Big)^2} \\
&= \sqrt{\sum_{s,a,s'\in S\times A\times S}P(s'|s,a)^2(\pi(a|s)-\bar{\pi}(a|s))^2} \\
&= \sqrt{\sum_{s,a\in S\times A}\Big(\sum_{s'\in S}P(s'|s,a)^2\Big)(\pi(a|s)-\bar{\pi}(a|s))^2} \\
&\leq \sqrt{\sum_{s,s'\in S\times S}(\pi(a|s)-\bar{\pi}(a|s))^2} = \|\boldsymbol{\pi}-\bar{\boldsymbol{\pi}}\|_2.
\end{aligned}$$

Where (i) comes from the fact that the Frobenius norm upper bounds the spectral norm, (ii) is by the definition of the Frobenius norm, and (iii) just plugs in the definition of the closed loop transition kernel (def 2.1.4) from there we can just rearange and isolate the $\left(\sum_{s'\in S}P(s'|s,a)^2\right)$ term, which since $P(\cdot,s,a)\in\Delta_S$ we know is less than 1. From there we just observe that we have gotten to the definition of the $l2$ norm.

Putting everything back together we have:

$$\begin{aligned}
\|\boldsymbol{\mu}^{\pi}-\boldsymbol{\mu}^{\bar{\pi}}\|_2 &\leq \|\boldsymbol{\pi}-\bar{\boldsymbol{\pi}}\|_2+\|\boldsymbol{\mu}_s-\bar{\boldsymbol{\mu}}_s\|_2 \\
&\leq \|\boldsymbol{\pi}-\bar{\boldsymbol{\pi}}\|_2+\frac{\gamma}{1-\gamma}\left\|P^{\boldsymbol{\pi}}-P^{\bar{\boldsymbol{\pi}}}\right\| \\
&\leq \|\boldsymbol{\pi}-\bar{\boldsymbol{\pi}}\|_2+\frac{\gamma}{1-\gamma}\|\boldsymbol{\pi}-\bar{\boldsymbol{\pi}}\|_2 \\
&= \frac{1}{1-\gamma}\|\boldsymbol{\pi}-\bar{\boldsymbol{\pi}}\|_2
\end{aligned}$$

D.4. Proof of proposition D.2.4 (Sufficient decrease)

Proof. Bubeck 2015 Starting from the definition of smoothness the proof follows naturally:

$$\begin{aligned}
D(\boldsymbol{z}^{(t+1)}) &\leq D(\boldsymbol{z}^{(t)})+\langle\nabla_z(\boldsymbol{z}^{(t)}),\boldsymbol{z}^{(t+1)}-\boldsymbol{z}^{(t)}\rangle-\frac{L_z}{2}\|\boldsymbol{z}^{(t+1)}-\boldsymbol{z}^{(t)}\|_2^2 && \text{(D.82)}\\
&= D(\boldsymbol{z}^{(t)})-\frac{1}{\eta_z}\langle\boldsymbol{z}^{(t+1/2)}-\boldsymbol{z}^{(t)},\boldsymbol{z}^{(t)}-\boldsymbol{z}^{(t+1)}\rangle-\frac{L_z}{2}\|\boldsymbol{z}^{(t+1)}-\boldsymbol{z}^{(t)}\|_2^2 && \text{(D.83)}\\
&\overset{(i)}{\leq} D(\boldsymbol{z}^{(t)})-\frac{L_z}{2}\big(\|\boldsymbol{z}^{(t+1/2)}-\boldsymbol{z}^{(t)}\|_2^2+\|\boldsymbol{z}^{(t)}-\boldsymbol{z}^{(t+1)}\|_2^2-\|\boldsymbol{z}^{(t+1)}-\boldsymbol{z}^{(t+1/2)}\|_2^2\big) && \text{(D.84)}\\
&\quad-\frac{L_z}{2}\|\boldsymbol{z}^{(t+1)}-\boldsymbol{z}^{(t)}\|_2^2 \\
&= D(\boldsymbol{z}^{(t)})-\frac{L_z}{2}\|\boldsymbol{z}^{(t)}-\boldsymbol{z}^{(t+1)}\|_2^2+\frac{L_z}{2}\|\boldsymbol{z}^{(t+1)}-\boldsymbol{z}^{(t+1/2)}\|_2^2 && \text{(D.85)}\\
&= D(\boldsymbol{z}^{(t)})-\frac{L_z}{2\eta_z^2}\|\nabla_z D(\boldsymbol{z}^{(t)})\|_2^2+\frac{L_z}{2}\|\boldsymbol{z}^{(t+1)}-\boldsymbol{z}^{(t+1/2)}\|_2^2\big) && \text{(D.86)}\\
&\overset{(ii)}{\leq} D(\boldsymbol{z}^{(t)})-\frac{1}{2L_z}\|\nabla_z D(\boldsymbol{z}^{(t)})\|_2^2+\frac{L_z}{2}\|\boldsymbol{z}^{(t+1)}-\boldsymbol{z}^{(t+1/2)}\|_2^2\big). && \text{(D.87)}
\end{aligned}$$

Where (i) and (i) hold because $\eta_z \leq \frac{1}{L_z}$. The proof is completed by rearranging and observing that the minizer $D(\boldsymbol{z}^*)$ is by definition smaller or equal to any other point $\boldsymbol{z}^{(t)} \in \text{dom}(\boldsymbol{z})$. □