Contextual Bilevel Reinforcement Learning for Incentive Alignment

Vinzenz Thoma *
ETH AI Center
vinzenz.thoma@ai.ethz.ch

Barna Pasztor *
ETH AI Center
barna.pasztor@ai.ethz.ch

Andreas Krause
ETH Zurich
krausea@ethz.ch

Giorgia Ramponi
University of Zurich
giorgia.ramponi@uzh.ch

Yifan Hu
EPFL & ETH Zurich
yifan.hu@epfl.ch

Abstract

The optimal policy in various real-world strategic decision-making problems depends both on the environmental configuration and exogenous events. For these settings, we introduce *Contextual Bilevel Reinforcement Learning* (CB-RL), a stochastic bilevel decision-making model, where the lower level consists of solving a contextual Markov Decision Process (CMDP). CB-RL can be viewed as a Stackelberg Game where the leader and a random context beyond the leader's control together decide the setup of many MDPs that potentially multiple followers best respond to. This framework extends beyond traditional bilevel optimization and finds relevance in diverse fields such as RLHF, tax design, reward shaping, contract theory and mechanism design. We propose a stochastic *Hyper Policy Gradient Descent (HPGD)* algorithm to solve CB-RL, and demonstrate its convergence. Notably, HPGD uses stochastic hypergradient estimates, based on observations of the followers' trajectories. Therefore, it allows followers to use any training procedure and the leader to be agnostic of the specific algorithm, which aligns with various real-world scenarios. We further consider the setting when the leader can influence the training of followers and propose an accelerated algorithm. We empirically demonstrate the performance of our algorithm for reward shaping and tax design.

1 Introduction

In Reinforcement Learning (RL), Markov Decision Processes (MDPs) [53] provide a versatile framework for capturing sequential decision-making problems across various domains such as health care [76], energy systems [52], economics [14], and finance [33]. A considerable amount of work has been devoted to solving standard MDPs [2, 62, 67]. However, in many applications, MDPs can be configured on purpose or affected by exogenous events, both of which can significantly impact the corresponding optimal policies. For example, in a simplified economic framework, the optimal decision of an individual household depends both on public policies and economic uncertainties [19, 34]. The policy maker in turn has to make decisions, anticipating the best response of differently-minded individual agents to its policies and exogenous events outside the policy maker's control.

To study such problems, we introduce Contextual Bilevel Reinforcement Learning (CB-RL), a hierarchical decision-making framework where followers solve a contextual Markov decision

*Equal Contribution

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

processes (CMDP) [32], configured by the leader. CB-RL is formalized as:

$$\begin{aligned} \min_{x} \quad & F(x) := \mathbb{E}_{\xi}[f(x, \pi^*_{x,\xi}, \xi)] \qquad \text{(leader, upper-level)} \\ \text{where} \quad & \pi^*_{x,\xi} = \operatorname*{argmax}_{\pi} J_{\lambda,x,\xi}(\pi) \quad \text{(follower, lower-level)}, \end{aligned} \tag{1}$$

where x represents the model configuration of the CMDP chosen by the leader, ξ represents the context that followers encounter, and the function $J_{\lambda,x,\xi}$ denotes the (entropy-regularized) reward of the CMDP for a policy π, a given model parameters x, a context ξ, and a regularization parameter λ.

In our framework, the leader chooses x to configure various aspects of the CMDP, such as state transitions, the initial state distribution, and the followers' reward functions. Modeling the problem as a CMDP instead of a standard MDP is essential when modeling situations where the environment is influenced by side information or personal perferences. For example, the context ξ can capture a wide range of real-world scenarios such as: (1) there is one follower, who responds optimally not only to the leader's chosen x but also to a side information ξ, such as weather or season, (2) there are multiple followers, each aiming to maximize their own utility, in which case ξ represents different possible follower preferences, and (3) there are multiple followers, each facing an uncertain contextual variable, i.e., $\xi = (i, \eta)$ represents the i-th follower encountering a specific context $\eta \sim \mathbb{P}_{\eta}$.

The proposed framework extends the concept of contextual bilevel optimization [37], where the follower engages in static contextual stochastic optimization rather than sequential decision-making. It also expands upon traditional MDP model design [78, 15] and configurable MDPs [50, 55], where typically only one follower attempts to solve an MDP, as opposed to CMDPs. We defer a full discussion of the related works to Appendix A. Beyond these fields, our framework finds various applications in Principal-Agent problems [8], RLHF [13, 58], dynamic Stackelberg games [27, 65], Security Games [60, 46], dynamic mechanism design [18], contract theory [41, 69], and tax design [19, 82, 34]. See Appendix B for the concrete CB-RL formulations of these applications.

Despite its wide applicability, to the best of our knowledge, there are no algorithms specifically designed for CB-RL. The closest is an algorithm from model design for MDPs [15], which can be adapted to our setting after some modifications. However, [15] requires the follower to solve the MDP deterministically using soft value iteration. Moreover, the full hypergradient is computed in each iteration, as part of which the leader requires access to the lower-level computations. Both of these aspects significantly restrict how well the method can scale to larger settings.

Instead, in this work, we propose a stochastic *Hyper Policy Gradient Descent (HPGD)* algorithm for the leader that solely relies on trajectory data from followers. The followers can use a variety of possibly stochastic learning algorithms to find an approximately optimal policy for the lower-level CMDPs. The leader in turn is agnostic of the exact algorithm used, as the hypergradient is estimated using only trajectory samples generated from the follower policy. The fact that both the lower-level and the hypergradient computation are stochastic makes HPGD salable to large problem settings.

We show non-asymptotic convergence of HPGD to a stationary point and validate these findings through experimental evidence. In scenarios where followers grant the leader full control over their training procedure, as posited in prior work [15], we present an accelerated HPGD algorithm, designed to minimize the number of lower-level iterations.

Our Contributions

- We introduce *Contextual Bilevel Reinforcement Learning (CB-RL)* that captures a wide range of important applications (Sec. 2). It is the first bilevel reinforcement learning framework that through the context ξ allows multiple followers and side information. We summarize the key differences to the previous literature in Table 1.
- We propose a stochastic *Hyper Policy Gradient Descent (HPGD)* algorithm that performs stochastic gradient descent on the upper-level objective (Sec. 3.2). Importantly, we are the first to estimate the hypergradient from lower-level trajectory samples instead of computing it exactly, while further providing convergence guarantees. Furthermore, our approach *is agnostic of the learning dynamics of the agent*, enabling followers to utilize a wide range of algorithms to solve the lower-level CMDPs. We only assume the leader can sample lower-level trajectories from an inexact oracle. For several widely-used RL algorithms, we explicitly show how to use them to build the inexact oracle needed by HPGD. Noteably, we are the first to consider stochastic lower-level learning algorithms, such as soft Q-learning.

Table 1: Summary of Related Works in Bilevel Reinforcement Learning.

	Context		r_x	P_x	μ_x	Agnostic	Deter	Upper	Lower	
	Multi	Side Info				Control	Stoch	Iterations	Iterations	Method
[15]			✓	✓		Control	Deter	$\mathcal{O}(\delta^{-2})$*	$\mathcal{O}(\log(\delta^{-1}))$	Soft-VI
[13]			✓			Agnostic	Deter	$\mathcal{O}(\delta^{-2})$	$\mathcal{O}(\log(\delta^{-1}))$	PG
[58]			✓	✓		Agnostic	Deter	$\mathcal{O}(\delta^{-2})$	$\mathcal{O}(\log(\delta^{-1}))$	PMG
[75]			✓			Agnostic	Deter	$\mathcal{O}(\delta^{-2})$	$\mathcal{O}(\log(\delta^{-1}))$	PMG
HPGD	✓	✓	✓	✓	✓	Agnostic	Stoch	$\mathcal{O}(\delta^{-4})$	$\mathcal{O}(\log(\delta^{-1}))$	Soft-VI
									$\mathcal{O}(\log(\delta^{-1}))$	NPG
									$\tilde{\mathcal{O}}(\delta^{-2})$	Soft-Q
HPGD	✓	✓	✓	✓	✓	Control	Stoch	$\mathcal{O}(\delta^{-4})$	$\mathcal{O}(\log(\delta^{-1}))$	RT-Q

Multi: Multiple followers. Side Info: Side information. Context: CMDP instead of MDP. r_x, P_x, and μ_x denote the dependence of rewards, transitions, and initial state distribution on x. Agnostic vs. Control: Whether leader can influence the training of the follower(s). Deter vs. Stoch: Requiring full knowledge of hypergradient or estimating it from samples. Complexity is based on $\|\nabla F(x)\|^2 \leq \delta^2$ instead of $\|\nabla F(x)\|^2 \leq \delta$. *[15] assumes convexity of F in x and considers $F(x) - \min F(x) \leq \delta$. VI: Value Iteration. PMG: Policy Mirror Gradient. PI: Policy Iteration. NPG: Natural Policy Gradient. Q: Q-learning. RT-Q: Randomly Truncated Soft Q-learning.

- We establish the non-asymptotic convergence rate of HPGD to a stationary point of the overall objective—despite the nonconvex lower-level problem (Sec. 3.2). When assuming full hypergradient information, i.e., deterministic updates, the outer iteration complexity of HPGD reduces to $\mathcal{O}(\delta^{-2})$, recovering previous results. Moreover, we discuss how to estimate the hypergradient if the upper-level loss function admits a specific form of cumulative costs (Sec. 3.3).
- When the leader is allowed to control the follwers' learning dynamics (Sec 4), we propose a stochastic accelerated algorithm denoted as HPGD RT-Q (Alg 8). It greatly reduces the number of lower-level soft Q-learning iterations from $\tilde{\mathcal{O}}(\delta^{-2})$ to $\mathcal{O}(\log(\delta^{-1}))$, such that we recover the rate for deterministic lower-level updates. For this result we leverage several techniques, such as mini-batches, multilevel Monte Carlo [29, 37], and importance sampling.
- We demonstrate the performance of HPGD for principal agent reward shaping and tax design (Sec. 5). In certain cases, the stochastic updates of HPGD are beneficial as they avoid local minima. Moreover, HPGD needs fewer outer iterations compared to benchmark algorithms.

2 Problem Formulation

We consider a bilevel optimization problem, where the follower solves a Contextual Markov Decision Process (CMPD) and the leader controls the configuration of the CMDP. In particular, the leader chooses a parameter $x \in X \subseteq \mathbb{R}^d$ and nature chooses a random context ξ according to a distribution $\mathbb{P}_\xi$. Together (x, ξ) parameterizes an MDP $\mathcal{M}_{x,\xi}$, which the follower aims to solve. $\mathcal{M}_{x,\xi}$ is defined by a tuple $(\mathcal{S}, \mathcal{A}, r_{x,\xi}, P_{x,\xi}, \mu_{x,\xi}, \gamma)$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $r_{x,\xi}(\cdot,\cdot) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $P_{x,\xi}(\cdot;\cdot,\cdot) : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0,1]$ denotes the transition kernel, $\mu_{x,\xi}$ indicates the initial state distribution, and γ is the discount factor. The subscript x, ξ implies that rewards, transitions, and initial state distribution depend on the leader's decision x and the context ξ. Connecting to previous works, for a fixed x, $\mathcal{M}_{x,\xi}$ is a *contextual MDP* [32] with respect to ξ. For a fixed ξ, $\mathcal{M}_{x,\xi}$ generalizes a *configurable MDP* [50]. Given $\mathcal{M}_{x,\xi}$, the follower maximizes an entropy-regularized objective by choosing a policy $\pi_{x,\xi}$, where $\pi_{x,\xi}(a;s)$ denotes the probability of choosing action a in state s.

$$\max_\pi J_{\lambda,x,\xi}(\pi) = \mathbb{E}_{s_0}\left[V^\pi_{\lambda,x,\xi}(s_0)\right] = \mathbb{E}_{s_0}\left[\mathbb{E}^\pi_{P_{x,\xi}}\left[\sum_{t=0}^{\infty} \gamma^t \left(r_{x,\xi}(s_t,a_t) + \lambda H(\pi;s_t)\right)\right]\right], \quad (2)$$

where $s_0 \sim \mu_{x,\xi}$, $a_t \sim \pi(\cdot;s_t)$, $s_{t+1} \sim P_{x,\xi}(\cdot;s_t,a_t)$ and $H(\pi;s) = -\sum_a \pi(a;s)\log\pi(a;s)$. We call $\lambda > 0$ the regularization parameter and $V^\pi_{\lambda,x,\xi}$ the value function. As standard in RL literature,

we define the related Q and advantage functions as:

$$Q^{\pi}_{\lambda,x,\xi}(s,a) = r_{x,\xi}(s,a) + \gamma \mathbb{E}_{s' \sim P_{x,\xi}(\cdot;s,a)} \left[V^{\pi}_{\lambda,x,\xi}(s') \right]$$
$$A^{\pi}_{\lambda,x,\xi}(s,a) = Q^{\pi}_{\lambda,x,\xi}(s,a) - V^{\pi}_{\lambda,x,\xi}(s) = Q^{\pi}_{\lambda,x,\xi}(s,a) - \sum_{a'} \pi(a';s) Q^{\pi}_{\lambda,x,\xi}(s,a'). \tag{3}$$

The unique optimal policy for (2) is given by $\pi^*_{x,\xi}(s;a) \propto \exp(Q^*_{\lambda,x,\xi}(s,a)/\lambda)$, i.e., the softmax of the optimal Q-function [51].[2] Given $x, \pi^*_{x,\xi}, \xi$, the leader in turn incurs a loss $f(x, \pi^*_{x,\xi}, \xi) \in \mathbb{R}$, which it wants to minimize in expectation over $\mathbb{P}_\xi$. To do so, it needs to choose x to align the follower's policy $\pi^*_{x,\xi}$ with the leader's objective. CB-RL can thus be formulated as the following stochastic bilevel optimization problem:

$$\begin{aligned} \min_{x} \quad & F(x) := \mathbb{E}_\xi[f(x, \pi^*_{x,\xi}, \xi)] \qquad \text{(leader, upper-level)} \\ \text{where} \quad & \pi^*_{x,\xi} = \operatorname*{argmax}_{\pi} J_{\lambda,x,\xi}(\pi). \quad \text{(follower, lower-level)} \end{aligned} \tag{4}$$

Equation (4) is well-defined due to entropy regularization ensuring the uniqueness of $\pi^*_{x,\xi}$. It further ensures $\pi^*_{x,\xi}$ is differentiable, stabilizes learning and appears in previous works [15]. Moreover, the difference between the regularized and unregularized problem generally vanishes as $\lambda \to 0$ [15, 26].

CB-RL captures many important real-world problems, including RLHF, dynamic mechanism design, tax design, and general principal-agent problems. We discuss these in detail in Appendix B.

3 Hyper Policy Gradient Descent Algorithm for CB-RL

In this section, we derive a simple expression for the hypergradient of CB-RL. We present HPGD and prove non-asymptotic convergence. HPGD can be combined with a large class of lower-level MDP solvers satisfying a mild inexact oracle assumption. We show this is the case for several popular RL algorithms. Furthermore, we present results for two important special cases of our problem: (1) when the upper-level objective decomposes as a discounted sum of rewards over the lower-level trajectories, and (2) when the leader can direct the lower-level algorithm. We defer all proofs to Appendix E. and make the following standard assumptions on how x and ξ influence the setup of the CMDP.

Assumption 3.1. *We assume the following conditions:*

- *f is L_f-Lipschitz continuous and S_f-smooth in x and π, uniformly for all ξ, i.e.*

$$\begin{aligned} \|f(x_1,\pi_1,\xi) - f(x_2,\pi_2,\xi)\|_\infty &\le L_f \left(\|x_1 - x_2\|_\infty + \|\pi_1 - \pi_2\|_\infty \right) \\ \|\partial_x f(x_1,\pi_1,\xi) - \partial_x f(x_2,\pi_2,\xi)\|_\infty &\le S_f \left(\|x_1 - x_2\|_\infty + \|\pi_1 - \pi_2\|_\infty \right) \\ \|\partial_\pi f(x_1,\pi_1,\xi) - \partial_\pi f(x_2,\pi_2,\xi)\|_\infty &\le S_f \left(\|x_1 - x_2\|_\infty + \|\pi_1 - \pi_2\|_\infty \right) \end{aligned}$$

- $\forall x, \xi : |r_{x,\xi}(s,a)| < \overline{R}$, $\|\partial_x \log P_{x,\xi}(s';s,a)\|_\infty < K_1$, $\|\partial_x r_{x,\xi}(s,a)\|_\infty < K_2$.

3.1 Hypergradient derivation

The leader's loss f depends on x directly and indirectly through $\pi^*_{x,\xi}$. Therefore, the derivative of f with respect to x is commonly referred to as the *hypergradient* to highlight this nested dependency. It is possible to obtain a closed-form expression of the hypergradient, using the implicit function theorem [28]. However, this involves computing and inverting the Hessian of the follower's value function, which can be computationally expensive and unstable [24, 47]. Instead, we leverage the fact that $\pi^*_{x,\xi}$ is a softmax function to explicitly compute its derivative with respect to x, which is given by $\frac{d\pi^*_{x,\xi}(s,a)}{dx} = \frac{1}{\lambda}\pi^*_{x,\xi}(a;s)\partial_x A^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a)$ [15], where $\partial_x A = (\partial_{x_1} A, \ldots, \partial_{x_d} A)$. Applying the Dominated Convergence Theorem to switch derivative and expectation, we arrive at Theorem 1.

Theorem 1. *Under Assumption 3.1, F is differentiable and the hypergradient is given by*

$$\frac{dF(x)}{dx} = \mathbb{E}_\xi \left[\frac{\partial_1 f(x, \pi^*_{x,\xi}, \xi)}{\partial x} + \mathbb{E}_{s\sim\nu, a\sim\pi^*_{x,\xi}} \left[\frac{1}{\lambda \nu(s)} \frac{\partial_2 f(x, \pi^*_{x,\xi}, \xi)}{\partial \pi^*_{x,\xi}(a;s)} \partial_x A^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a) \right] \right], \tag{5}$$

where ν is any sampling distribution with full support on the state space $\mathcal{S}$.

[2] For brevity, we notationally drop the dependence of $\pi_{x,\xi}$ on λ.

Algorithm 1 Hyper Policy Gradient Descent (HPGD)

Input: Iterations T, Learning rate α, Regularization λ, Trajectory oracle o, Initial point x_0
for $t = 0$ to $T-1$ **do**
  $\xi \sim \mathbb{P}_\xi$, $s \sim \nu$ and $a \sim \pi^o_{x,\xi}(\cdot; s)$
  $\widehat{\partial_x A^{\pi^o_{x,\xi}}_{\lambda,x,\xi}}(s,a) \leftarrow$ `GradientEstimator`(ξ, x_t, s, a, o) (Algorithm 2)
  $\widehat{\frac{dF}{dx}} \leftarrow \frac{\partial_1 f(x_t, \pi^o_{x_t,\xi}, \xi)}{\partial x} + \frac{1}{\lambda \nu(s)} \frac{\partial_2 f(x_t, \pi^o_{x_t,\xi}, \xi))}{\partial \pi(s,a)} \widehat{\partial_x A^{\pi^o_{x,\xi}}_{\lambda,x,\xi}}(s,a)$
  $x_{t+1} \leftarrow x_t - \alpha \widehat{\frac{dF}{dx}}$
end for
Output: $\hat{x}_T \sim \text{Uniform}(\{x_0, \ldots, x_{T-1}\})$

The first term captures the direct influence of x on f, and the second the indirect influence through $\pi^*_{x,\xi}$. For now we assume the leader knows $\partial_1 f(\cdot, \pi, \xi)$ and $\partial_2 f(x, \cdot, \xi)$. It remains to compute $\partial_x A^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a)$, i.e. the partial derivative with respect to x evaluated for a given policy. For this, cf. (3), we need $\partial_x Q^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a)$. We derive an expression for the latter in Theorem 2. The proof adapts the analysis of the policy gradient theorem to account for the dependence of $P_{x,\xi}, \mu_{x,\xi}$ and $r_{x,\xi}$ on x.

Theorem 2. *For given π, x, ξ, it holds that:*

$$\partial_x Q^\pi_{\lambda,x,\xi}(s_0,a_0) = \mathbb{E}^\pi_{s,a}\left[\sum_{t=0}^{\infty} \gamma^t \frac{dr_{x,\xi}(s_t,a_t)}{dx} + \gamma^{t+1} \frac{d \log P_{x,\xi}(s_{t+1}; s_t, a_t)}{dx} V^\pi_{\lambda,x,\xi}(s_{t+1})\right].$$

Note, Theorems 1 and 2 generalize existing results in model design for MDPs to CMDPs [15, 78].

3.2 HPGD Algorithm and Convergence Analysis

Computing the exact hypergradient is computationally expensive and thus infeasible in larger settings. Instead, to minimize $F(x)$, one would ideally sample unbiased estimates of the hypergradient in Equation (5) and run stochastic gradient descent (SGD). However, the leader does not have access to $\pi^*_{x,\xi}$ and generally no control over the training procedure of the lower level. Instead, we assume the follower adapts any preferred algorithm to solve the MDP up to a certain precision δ and the leader can observe trajectories from the follower's policy. Such a setting is well-motivated by economic applications.

Assumption 3.2. *For any $\mathcal{M}_{x,\xi}$, the leader has access to an oracle o, which returns trajectories sampled from a policy $\pi^o_{x,\xi}$ such that $\forall x, \forall \xi : \mathbb{E}_o\left[\left\|\pi^*_{x,\xi} - \pi^o_{x,\xi}\right\|^2_\infty\right] \leq \delta^2$.*

We will show that Assumption 3.2 is relatively mild and holds for a variety of RL algorithms. Given access to trajectories generated by $\pi^o_{x,\xi}$, the leader can construct an estimator of $\partial_x A^{\pi^o_{x,\xi}}_{\lambda,x,\xi}(s,a)$ by rolling out $\pi^o_{x,\xi}$ for T steps, where $T \sim \text{Geo}(1-\gamma)$. We defer the construction (Algorithm 2) and proof of unbiasedness (Proposition 3) to the Appendix. Using this estimator, we introduce HPGD in Algorithm 1. As F is generally nonconvex due to the bilevel structure [28], we demonstrate non-asymptotic convergence to a stationary point of F, which matches the lower bound for solving stochastic smooth nonconvex optimization [1].

Theorem 3. *Using HPGD, under Assumptions 3.1 and 3.2, for $\alpha \leq 1/(2S_f)$, we have the following:*

$$\mathbb{E}\left\|\frac{dF(\hat{x}_T)}{dx}\right\|^2 = \mathcal{O}\Big(\frac{1}{\alpha T} + \delta + \alpha\Big). \tag{6}$$

For $\alpha = \mathcal{O}(1/\sqrt{T})$ and $\delta = \mathcal{O}(1/\sqrt{T})$, HPGD converges to a stationary point at rate $\mathcal{O}(1/\sqrt{T})$.

Proof sketch. Using the smoothness of F and the fact that $\hat{x}_T$ is uniformly sampled from all iterates, we upper bound the left side of (6) by the sum of three terms. The first is $|F(x_0) - \min_x F(x)|/(\alpha T)$. The second depends on the bias of our gradient estimate, which we show is linear in δ. The last term depends on α times the variance of our estimator, which is bounded. □

To the best of our knowledge, Theorem 3 is the first result that shows convergence when using stochastic estimates for the hypergradient. When the hypergradient can be computed exactly, the last $\mathcal{O}(\alpha)$ term vanishes and we recover the deterministic convergence rates of previous works [15, 13, 58].

Another major advantage of HPGD is that the follower can use any (possibly stochastic) algorithm satisfying Assumption 3.2 to solve the lower-level MDP, while the leader only needs access to generated trajectories. While Assumption 3.2 certainly holds if the follower solves the MDP exactly, for example with an LP-solver, we are interested in verifying it for common RL algorithms, which can scale to larger state and action spaces. In Appendix E.8, we prove non-asymptotic convergence to $\pi^*_{x,\xi}$ for Soft Value Iteration, which converges at rate $\mathcal{O}(\log 1/\delta)$ (Proposition 5); Q-learning, which converges at rate of $\mathcal{O}(\log(1/\delta)/\delta^2)$ (Proposition 6) and Natural Policy Gradient, which converges at rate of $\mathcal{O}(\log 1/\delta)$ (Proposition 8). Additionaly, we show Vanilla Policy Gradient converges asymptotically in Proposition 7. All these Algorithms thus satisfy Assumption 3.2, which makes HPGD scalable and widely applicable to settings where followers might use a variety of model-free or model-based algorithms.

3.3 Upper-Level Discounted Reward Objective

So far we assumed the leader knows $\partial_1 f(\cdot, \pi, \xi)$ and $\partial_2 f(x, \cdot, \xi)$. In this subsection, instead, we assume f can be written as the negative expected sum of discounted rewards over the lower-level trajectories and show how to estimate the hypergradient from trajectory samples without explicit knowledge of $\partial_1 f(\cdot, \pi, \xi)$ and $\partial_2 f(x, \cdot, \xi)$. In many practical applications, such as reward shaping, or dynamic mechanism design (cf. Appendix B), the loss f satisfies:

$$f(x, \pi^*_{x;\xi}, \xi) = -\mathbb{E}^{\pi^*_{x,\xi}}_{s_0 \sim \mu}\Big[\sum\nolimits_t \gamma^t \overline{r}_{x,\xi}(s_t, a_t)\Big]. \tag{7}$$

Here $\overline{r}_{x,\xi}$ represents the reward of the leader, which is generally distinct from the follower's reward. The expectation is taken over trajectories induced by the lower-level $\pi^*_{x,\xi}$. In this case, the leader does not know the partial derivatives of f but can still estimate the hypergradient from trajectory samples. The following proposition follows from a similar anlysis as the policy gradient theorem.

Proposition 1. *If f decomposes as in Equation (7), then $\frac{dF(x)}{dx}$ can be expressed as follows:*

$$\frac{dF(x)}{dx} = \mathbb{E}_\xi\left[\mathbb{E}^{\pi^*_{x,\xi}}_{s_0 \sim \mu_{x,\xi}}\left[\sum_{t=0}^{\infty} \gamma^t \left(\frac{1}{\lambda}\partial_x A^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s_t, a_t)\overline{Q}_{x,\xi}(s_t, a_t) + \frac{d\overline{r}_{x,\xi}(s_t, a_t)}{dx} + \partial_x \log P_{x,\xi}(s_t; s_{t-1}, a_{t-1})\overline{V}_{x,\xi}(s_t)\right)\right]\right], \tag{8}$$

where for compactness, we slightly abuse notation to express $\mu_{x,\xi}(s_0)$ as $P_{x,\xi}(s_0, a_{-1}, s_{-1})$.

Here $\overline{V}_{x,\xi}, \overline{Q}_{x,\xi}$ are the (unregularized) value and state action value functions with respect to $\overline{r}_{x,\xi}$. Comparing to Theorem 1, note that the expectation is over trajectories with starting states distributed according to the actual initial distribution $\mu_{x,\xi}$ instead of some ν. We discuss how to construct estimators for (8) in Algorithm 5 (Appendix D) and prove unbiasedness in Proposition 4 (Appendix E.7). A special case of Equation (8) appeared in [15], where they consider model design for MDPs, that does not take into account contextual uncertainty or the possibility of multiple followers, i.e when the support of ξ is a singleton.

4 Accelerated HPGD with Full Lower-Level Access

Previously, we assumed that the leader does not know the solver used in the lower level and queries trajectories from an oracle. However, in certain settings, such as model design [15], and dynamic mechanism design [18], the leader can additionally influence how the followers solve the CMDP. In this section, we focus on the case when the followers use a stochastic tranining procedure, which usually require a polynomial number of steps in terms of δ^{-1}, to learn the optimal lower-level policy. We argue that if the leader has influence on the followers' training procedure, we can greatly reduce the number of lower-level iterations.

Let us assume that the lower level is solved using vanilla soft Q-learning (Algorithm 4 in Appendix D). According to Proposition 6 (Appendix E.7), the follower needs to run $T = \mathcal{O}(K2^K)$ iterations

Table 2: Bias, variance and lower-level iteration complexity of hypergradient estimators when using vanilla soft Q-learning and RT-Q.

	Vanilla	RT-Q
Bias	$\mathcal{O}(2^{-K/2})$	$\mathcal{O}(2^{-K/2})$
Variance	$\mathcal{O}(1)$	$\mathcal{O}(K)$
Complexity	$\mathcal{O}(K2^K)$	$\mathcal{O}(K^2)$

to ensure that $\mathbb{E}\|\pi^T_{x,\xi} - \pi^*_{x,\xi}\|^2_\infty \leq 2^{-K}$ and thus $\left\|\mathbb{E}\left[\frac{dF(x)}{dx} - \widehat{\frac{dF_T}{dx}}\right]\right\|_\infty = \mathcal{O}(2^{-K/2})$, where π^T denotes the learned policy after running T-th Q-learning iterations and $\widehat{\frac{dF_T}{dx}}$ denotes the corresponding hypergradient estimator.

To reduce the lower-level iteration complexity, we propose a randomized early stopping scheme over the lower-level soft Q-learning iterations, denoted as randomly-truncated soft Q-learning (RT-Q). The pseudocode is given in Algorithm 8 (Appendix E.5). We illustrate the high-level idea below.

Without loss of generality, consider a subsequence $t_k := \mathcal{O}(k2^k)$ such that $t_K := T$. Let $\frac{d}{dx}F_T$ denote the hypergradient estimator, based on the T-th policy iterate π^T. It holds that:

$$\frac{d}{dx}F_T = \frac{d}{dx}F_{t_K} = \frac{d}{dx}F_{t_1} + \sum_{k=1}^{K-1}\left(\frac{d}{dx}F_{t_{k+1}} - \frac{d}{dx}F_{t_k}\right) = \frac{d}{dx}F_{t_1} + \mathbb{E}_{k\sim p_k}\left[\frac{\frac{d}{dx}F_{t_{k+1}} - \frac{d}{dx}F_{t_k}}{p_k}\right],$$

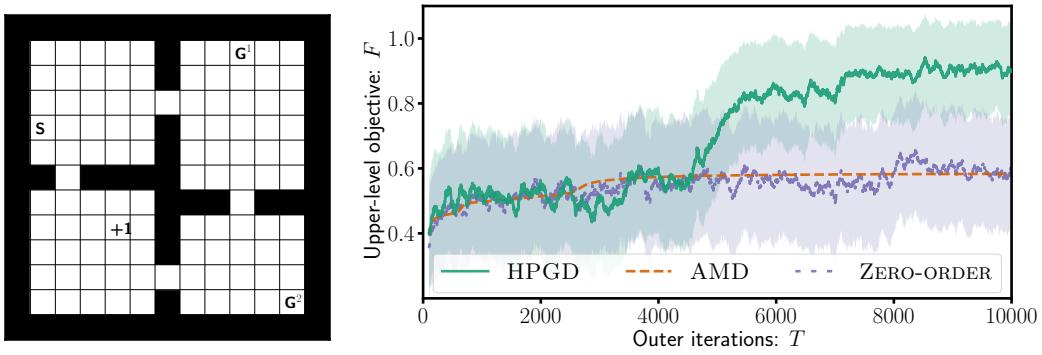
where p_k denotes a truncated geometric distribution, such that $p_k \propto 2^{-k}$. The above shows that $\frac{d}{dx}F_{t_1} + p_k^{-1}\left[\frac{d}{dx}F_{t_{k+1}} - \frac{d}{dx}F_{t_k}\right]$ with $k \sim p_k$ is an unbiased estimator of $\frac{d}{dx}F_T$. Using this estimator, the follower does not need to run $\mathcal{O}(K2^K)$ soft Q-learning iterations but in expectation only $\sum_{k=1}^{K-1} p_k t_k = \mathcal{O}(K^2)$ iterations. This implies that if the leader can direct how the followers learn and observe behaviors sampled from their learned policies, we can generate a hypergradient estimator with the same bias as $\frac{d}{dx}F_T$ but a much smaller lower-level iteration complexity. We formalize our results in the following Theorem.

Theorem 4 (Improved iteration complexity using RT-Q)**.** *Using Randomly Truncated soft Q-learning (RT-Q) instead of vanilla soft Q-learning to estimate the hypergradient, we achieve the bias, variance, and lower-level iteration complexity results summarized in Table 2.*

The idea has been previously studied for contextual bilevel optimization under the name randomly truncated multilevel Monte-Carlo [29, 36, 37]. The reduction in the iteration complexity generally comes at the expense of an increased variance of the hypergradient estimator. In [37], this increase is logarithmic as the lower-level problem is a static optimization problem and samples generated to estimate the hypergradient are independent from the lower-level decision variable. This structure is crucial for controlling the increased variance of the hypergradient estimator. However, for CB-RL, rollouts generated from $\pi^t_{x,\xi}$ are used to estimate the hypergradient. These trajectory samples thus depend on the lower-level decision and it is not possible to control the variance as in [37]. To address this issue, we notice that the major source of randomness in our hypergradient estimators stems from the estimator $\widehat{\partial_x A^{\pi^{t_k}}}(s,a)$ computed by `GradientEstimator` (Algorithm 2). We control this randomness by sampling multiple trajectories with a random length, which is in expectation $\mathcal{O}(1)$. We further sample an action a once from $\pi^{t_{k+1}}$ and then use it to compute both $\widehat{\partial_x A^{\pi^{t_{k+1}}}}(s,a)$ and $\widehat{\partial_x A^{\pi^{t_k}}}(s,a)$, using importance sampling. Combining all these tricks with multi-level Monte Carlo, RT-Q achieves a variance of $\mathcal{O}(K)$, where $K = \mathcal{O}(\log(\delta^{-1}))$.

5 Numerical Experiments

We illustrate the performance of HPGD in the Four-Rooms environment and on the Tax Design for Macroeconomic Model problem [34, 15] that we extend to multiple households with diverse preferences. We use Adaptive Model Design (AMD) [15] and a zeroth-order gradient approximation algorithm for comparison. Details on the zeroth-order algorithm are deferred to Appendix F.1.2. We note that AMD is not directly applicable to CB-RL as it was designed for solving an MDP without



(a) Four Room State Space

(b) Upper-level objective values, F, over the number of outer iterations

Figure 1: Four-Rooms State Space and Performance. **Left**: S denotes the start state while G^1 and G^2 denote goal states that are considered separate tasks. $+1$ denotes the target cell to which the upper-level aims to steer the lower-level MDP. **Right:** HPGD escapes local optima achieving higher performance than comparison algorithms.

context. We apply it with modifications described in Appendix F.1.1. To our knowledge, such a zeroth-order gradient method is also the first of its kind for CB-RL. The main distinction between the algorithms is the zeroth-order algorithm requires two oracle queries for each gradient calculation while HPGD and AMD require only one. However, the zeroth-order method only needs to observe the function value of the upper level while the latter two require first-order information about the lower-level contextual MDP. In particular, AMD assumes complete access to the MDP to calculate the exact hypergradient while HPGD relies only on trajectory samples. Technical details about the implementation [3] are deferred to Appendix (F.2).

5.1 Four-Rooms Environment

Figure (1a) depicts the lower-level CMDP for the Four-Rooms environment. S denotes the initial position while G^1 and G^2 are goal states. We consider the two goal states as separate tasks and define ξ in Equation (4) to be the uniform distribution over the set of tasks, i.e., $\xi \sim \text{Uniform}(\{1, 2\})$. We denote the goal state in each task by G^ξ. The state space $\mathcal{S}$ is defined by the cells of the grid world while the actions are the movements in the four directions. In each step t, with probability $2/3$, the agent moves to s_{t+1} following the chosen direction a_t while it takes a random movement with probability $1/3$. The reward is always zero except when $s_t = G^\xi$ where $r(s_t, a_t) = 1$, and the episode resets. To incentivize taking the shortest path, we set the discount factor as $\gamma = 0.99$.

For the upper level, we let x parameterize an additive penalty function $\tilde{r}_x : \mathcal{S} \times \mathcal{A} \to [-0.2, 0.0]$ [4], such that the follower receives a reward of $r + \tilde{r}_x$, as in the Principal-Agent problem [8]. The goal of the leader is to steer the followers through the cell marked with $+1$ in Figure 1a, denoted by s^{+1}, while keeping the penalties allocated to states to their minimum. We define $\overline{r}$ in Equation (7) as

$$\overline{r}_{x,\xi}(s_t, a_t) = \mathbb{1}_{\{s_t = s^{+1}\}} - \beta \mathbb{1}_{\{s_t = G^\xi\}} \sum\nolimits_{s,a} \tilde{r}_x(s, a),$$

where $\mathbb{1}$ is the indicator function and the second term defines the cost associated with implementing the penalties for the lower level. Note that there is a trade-off between the terms in $\overline{r}$ depending on the context variable ξ. If $\xi = 2$, the desired change in the follower's policy can be achieved with small interventions since the shortest path from S to G^2 is already going through the bottom-left room. When $\xi = 1$, the leader must completely block the shortest path from S to G^1 to divert the follower through the desired state. An efficient algorithm for this CB-RL problem therefore must avoid the local optimum of setting $\tilde{r} = 0$ and find the balance between the follower visiting state s^{+1} and implementing too large penalties in the CMDP.

Figure (1b) depicts the upper-level's objective function over the learning iterations t with hyperparameters $\lambda = 0.001$ and $\beta = 1.0$. HPGD outperforms both AMD and the Zero-Order algorithms

[3] We implemented our experiments end-to-end in JAX [10] for its runtime benefits and ease of experimentation. The code is available at https://github.com/lasgroup/HPGD.

[4] The parametrization of this function is described in Appendix F.2.1.

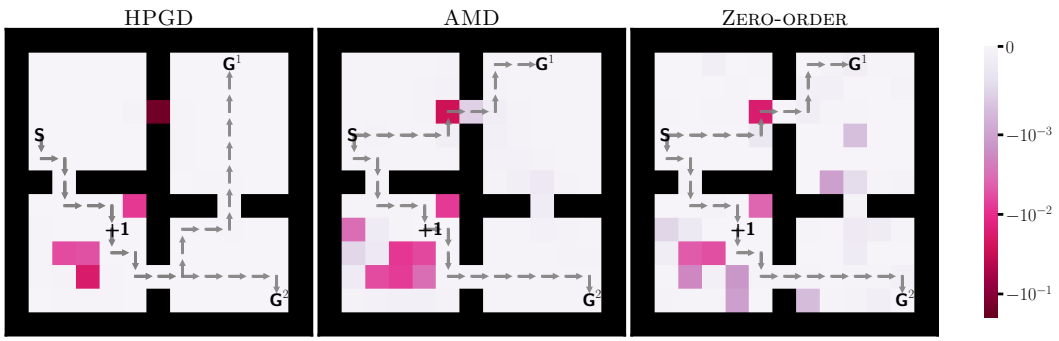


Figure 2: Reward penalties given to the lower-level agent in each state of the Four-Rooms problem optimized by the HPGD, AMD, and Zero-Order, respectively. HPGD efficiently steers the lower-level MDP when the task is to reach G^1 while others are only successful in the case of G^2.

Table 3: Performance over hyperparameters β and λ for the Four Rooms Problem averaged over 10 random seeds with standard errors. Algorithms perform on-par for most hyperparameters while HPGD outperforms others in few. AMD enjoys low variance due to the non-stochastic gradient updates while Zero-Order suffers from the most variation.

Parameters		Algorithms		
λ	β	HPGD	AMD	Zero-Order
0.001	1	**0.91** $\pm$ 0.088	0.58 $\pm$ 0.000	0.59 $\pm$ 0.059
0.001	3	0.51 $\pm$ 0.006	0.51 $\pm$ 0.000	0.50 $\pm$ 0.005
0.001	5	0.46 $\pm$ 0.006	0.46 $\pm$ 0.003	0.46 $\pm$ 0.007
0.003	1	0.95 $\pm$ 0.002	1.00 $\pm$ 0.000	0.91 $\pm$ 0.048
0.003	3	**0.73** $\pm$ 0.001	0.39 $\pm$ 0.000	0.40 $\pm$ 0.028
0.003	5	0.29 $\pm$ 0.003	0.32 $\pm$ 0.000	0.32 $\pm$ 0.002
0.005	1	1.17 $\pm$ 0.011	1.28 $\pm$ 0.003	1.15 $\pm$ 0.026
0.005	3	1.01 $\pm$ 0.002	1.13 $\pm$ 0.004	1.02 $\pm$ 0.027
0.005	5	0.87 $\pm$ 0.003	0.97 $\pm$ 0.009	0.79 $\pm$ 0.027

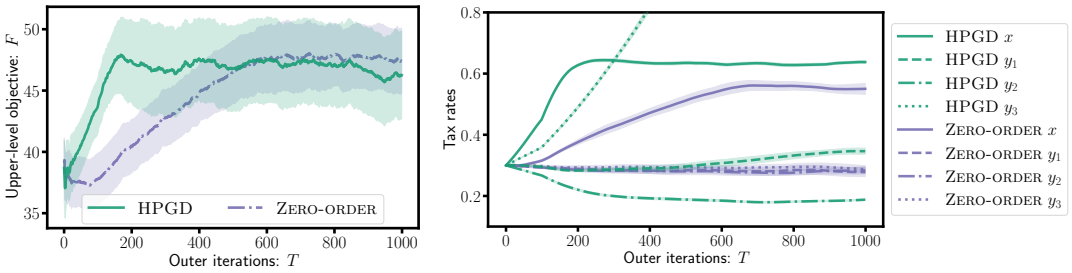
in this instance in terms of overall performance. The major difference in their performances is that HPGD successfully escapes the local optimum of $\tilde{r} = 0$ after about 5000 steps and assigns all the additive penalty budget to states in the gridworld. On the contrary, AMD and Zero-Order converge to the local optimum of minimizing the implementation penalty term in $\overline{r}$. They only utilize 38% and 26% of the available budget of -0.2 to divert the follower when $\xi = 2$ but neglect the goal state G^1.

Figure 2 shows the value of additive penalties $\tilde{r}$ in the state space with the highest probability paths for the goal states. HPGD successfully blocks the follower when $\xi = 1$ and diverts its shortest path from S to G^1 along the other rooms, while AMD and Zero-Order fail to assign sufficient penalty to the upper corridor to cause the same effect. All algorithms are successful in ensuring that the shortest path through the bottom-left room is going through the marked state.

The parameters λ and β were chosen for demonstration purposes to highlight the capability of HPGD to escape local minima, as has been observed for SGD [72]. However, we emphasize that in the majority of the cases, the three algorithms perform equally as shown in Table 3. We provide the figures for the remaining hyperparameters in Appendix F.2.3. The slightly higher performance of AMD and low standard error among initializations is expected since this algorithm calculates the gradient of f deterministically while HPGD and Zero-Order rely on stochastic estimates yielding more variations, especially for the Zero-Order approach.

5.2 Tax Design for Macroeconomic Models

We test HPGD on a bilevel macroeconomic model evaluating taxation schemes based on [34, 15]. The economic model consists of a finite set of consumption goods $i \in \{1, \dots, M\}$ each with price p_i. We choose $M = 3$ with unit prices. On the lower-level, at each time step t, s_t denotes the accumulated assets of a household which in turn chooses the number of hours worked n_t



(a) Performance on the Tax Design problem. HPGD quickly learn optimal tax policies while Zero-Order takes more iterations.

(b) Tax rates over the outer iterations. HPGD increases income tax quickly and distinguishes VAT rates according to preferences.

and consumption $c_{i,t}$. The environment updates the accumulated assets of the household as $s_{t+1} = s_t + (1-x)wn_t - \sum_{i=1}^{M} c_{i,t} + \epsilon$ where ϵ is sampled from a normal distribution with mean 0 and standard deviation ς. We clip the accumulated assets to the range $[-100, 100]$ for numerical stability. The utility of a household is given by $u_t = \sigma(s_t) - \theta n_t^2 + \prod_{i=1}^{M} (c_{i,t}/(p(1+y_i)))^{\alpha_i}$ where the product-of-consumption term corresponds to the Cobb-Douglas function [57] and $\sigma(\cdot)$ is the value of accumulated assets. We define $\mathbb{P}_\xi$ as the distribution of households representing different socio-economic groups in the economy. In particular, we define two equal-sized groups with $\alpha = (0.6, 0.3, 0.1)$ and $(0.1, 0.7, 0.2)$ that model their different preferences over the goods in the economy and optimize their consumption behavior using regularized deep Q-learning [26] with $\lambda = 0.2$. On the upper-level, a tax designer is optimizing the discounted sum of social welfare by setting the income tax $x \in [0, 3]$ and value-added tax $y_i \in [0, 3]$ for $i \in \{1, \dots, M\}$. In each time step t, the social welfare is defined as $v_t = \omega(s_t) + \sum_{i=1}^{M} c_{i,t}/(1+y_i) + \phi \log(\sum_{i=1}^{M} c_{i,t} y_i/(1+y_i) + wxn_t)$ where $\omega(\cdot)$ is the utility of accumulated assets and ϕ is a positive hyperparameter.

Figure 3a demonstrates that HPGD can successfully optimize the hyperparameters even in continuous complex problems. Benefiting from first-order information, HPGD converges faster than the Zero-Order algorithm and increases the social-welfare by about 25%. Additionally, HPGD shows better qualitative results for the optimised tax rates in Figure 3b. HPGD swiftly increases the income tax to improve its objective by increasing tax revenues while sets value-added tax rates according to the preferences of the household groups. Households on average prefer the second good the most, therefore, y_2 is set low to increase consumption and therefore maximize the second term in v_t while the third good is the least preferred for which the tax rate is increased[5] since consumption is already low. While Zero-Order shows equivalent performance on Figure 3a to HPGD it fails to distinguish goods when setting value-added tax rates. We defer further details on the implementation, hyperparameter choices, and additional results to Appendix F.3.1.

6 Conclusion

We introduce CB-RL, a class of stochastic bilevel optimization problems with lower-level contextual MDPs that capture a wide range of important applications, where a leader aims to design environments and incentive structures that align the followers' policies with the leader's upper-level objective. We propose HPGD, an oracle-based algorithmic framework, and analyze its convergence. Importantly, HPGD works with any existing algorithm that solves the lower-level CMDP to near-optimality, making it suitable in various regimes when the leader can only observe trajectories of followers. Moreover, HPGD is the first provably convergent algorithm in this area, which uses stochastic estimates of the hypergradient. We further propose RT-Q, a more efficient algorithm and study its bias, variance, and cost when the leader can fully control the followers' training. Numerical results further validate the expressiveness of the proposed model and the performance of our algorithm. Future directions include 1) applying HPGD in various real-world applications, 2) studying the setting when the lower-level problem is a game, and 3) studying single-loop algorithms for bilevel reinforcement learning with when the lower-level is just an MDP.

[5] $y_2 = 1.9$ after 1000 iterations. The figure is cropped for better readability.

Acknowledgements

The authors acknowledge constructive suggestions from Niao He and Sven Seuken. V. Thoma and B. Pasztor are supported by an ETH AI Center Doctoral Fellowship. V. Thoma acknowledges funding from the Swiss National Science Foundation (SNSF) Project Funding No. 200021-207343. This work was supported as a part of NCCR Automation, a National Centre of Competence (or Excellence) in Research, funded by the SNSF (grant number 51NF40_225155).

References

[1] Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *Mathematical Programming*, 199(1):165–214, 2023. (Cited on page 5.)

[2] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017. (Cited on page 1.)

[3] Kavosh Asadi and Michael L. Littman. An alternative softmax operator for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 243–252, Sydney, NSW, Australia, 2017. JMLR.org. (Cited on page 49.)

[4] Jan Balaguer, Raphael Koster, Christopher Summerfield, and Andrea Tacchetti. The good shepherd: An oracle agent for mechanism design. In *ICLR 2022 Workshop on Gamification and Multiagent Solutions*, 2022. (Cited on page 19.)

[5] Jonathan F Bard. *Practical bilevel optimization: algorithms and applications*, volume 30. Springer Science & Business Media, 2013. (Cited on page 19.)

[6] Tobias Baumann, Thore Graepel, and John Shawe-Taylor. Adaptive mechanism design: Learning to promote cooperation. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020. (Cited on page 19.)

[7] Jacob Beck, Risto Vuorio, Evan Zheran Liu, Zheng Xiong, Luisa Zintgraf, Chelsea Finn, and Shimon Whiteson. A survey of meta-reinforcement learning. *arXiv preprint arXiv:2301.08028*, 2023. (Cited on page 21.)

[8] Omer Ben-Porat, Yishay Mansour, Michal Moshkovitz, and Boaz Taitler. Principal-agent reward shaping in mdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9502–9510, 2024. (Cited on pages 2, 8, 20, and 21.)

[9] Mathieu Blondel and Vincent Roulet. The elements of differentiable programming, 2024. (Cited on page 46.)

[10] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. (Cited on page 8.)

[11] Seth Brown, Saumya Sinha, and Andrew J. Schaefer. Markov decision process design: A framework for integrating strategic and operational decisions. *Operations Research Letters*, 54:107090, May 2024. (Cited on page 19.)

[12] Shicong Cen, Chen Cheng, Yuxin Chen, Yuting Wei, and Yuejie Chi. Fast global convergence of natural policy gradient methods with entropy regularization. *Operations Research*, 70(4):2563–2578, July 2022. (Cited on page 52.)

[13] Souradip Chakraborty, Amrit Bedi, Alec Koppel, Huazheng Wang, Dinesh Manocha, Mengdi Wang, and Furong Huang. PARL: A unified framework for policy alignment in reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*, 2024. (Cited on pages 2, 3, 6, 19, and 20.)

[14] Arthur Charpentier, Romuald Elie, and Carl Remlinger. Reinforcement learning in economics and finance. *Computational Economics*, pages 1–38, 2021. (Cited on page 1.)

[15] Siyu Chen, Donglin Yang, Jiayang Li, Senmiao Wang, Zhuoran Yang, and Zhaoran Wang. Adaptive model design for markov decision process. In *International Conference on Machine Learning*, pages 3679–3700. PMLR, 2022. (Cited on pages 2, 3, 4, 5, 6, 7, 9, 17, 19, 20, 21, 46, and 52.)

[16] Tianyi Chen, Yuejiao Sun, and Wotao Yin. Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems. *Advances in Neural Information Processing Systems*, 34:25294–25307, 2021. (Cited on page 19.)

[17] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. (Cited on page 20.)

[18] Michael Curry, Vinzenz Thoma, Darshan Chakrabarti, Stephen McAleer, Christian Kroer, Tuomas Sandholm, Niao He, and Sven Seuken. Automated design of affine maximizer mechanisms in dynamic settings. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(9):9626–9635, March 2024. (Cited on pages 2, 6, 19, and 21.)

[19] Michael Curry, Alexander Trott, Soham Phade, Yu Bai, and Stephan Zheng. Learning solutions in large economic networks using deep multi-agent reinforcement learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, page 2760–2762, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems. (Cited on pages 1, 2, and 19.)

[20] Bo Dai, Albert Shaw, Lihong Li, Lin Xiao, Niao He, Zhen Liu, Jianshu Chen, and Le Song. SBEED: Convergent reinforcement learning with nonlinear function approximation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1125–1134. PMLR, 10–15 Jul 2018. (Cited on pages 19 and 48.)

[21] Stephan Dempe. *Foundations of bilevel programming*. Springer Science & Business Media, 2002. (Cited on page 19.)

[22] Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. (Cited on page 19.)

[23] Manfred Diaz, Charlie Gauthier, Glen Berseth, and Liam Paull. Generalization games for reinforcement learning. In *ICLR Workshop on Agent Learning in Open-Endedness*, 2022. (Cited on page 19.)

[24] Tanner Fiez, Benjamin Chasnov, and Lillian Ratliff. Implicit learning dynamics in stackelberg games: Equilibria characterization, convergence analysis, and empirical study. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3133–3144. PMLR, 13–18 Jul 2020. (Cited on pages 4 and 19.)

[25] Jakob Foerster, Richard Y. Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, page 122–130, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems. (Cited on page 19.)

[26] Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169. PMLR, 2019. (Cited on pages 4, 10, and 19.)

[27] Matthias Gerstgrasser and David C. Parkes. Oracles & followers: Stackelberg equilibria in deep multi-agent reinforcement learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the*

40th International Conference on Machine Learning, volume 202 of *Proceedings of Machine Learning Research*, pages 11213–11236. PMLR, 23–29 Jul 2023. (Cited on pages 2 and 19.)

[28] Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018. (Cited on pages 4, 5, 19, and 34.)

[29] Michael B. Giles. Multilevel monte carlo methods. *Acta Numerica*, 24:259–328, 2015. (Cited on pages 3 and 7.)

[30] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1352–1361, Sydney, NSW, Australia, 2017. JMLR.org. (Cited on page 49.)

[31] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart Russell, and Anca D. Dragan. Inverse reward design. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6768–6777, Red Hook, NY, USA, 2017. Curran Associates Inc. (Cited on page 19.)

[32] Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual markov decision processes. *arXiv preprint arXiv:1502.02259*, 2015. (Cited on pages 2 and 3.)

[33] Ben Hambly, Renyuan Xu, and Huining Yang. Recent advances in reinforcement learning in finance. *Mathematical Finance*, 33(3):437–503, 2023. (Cited on page 1.)

[34] Edward Hill, Marco Bardoscia, and Arthur Turrell. Solving heterogeneous general equilibrium economic models with deep reinforcement learning. *arXiv preprint arXiv:2103.16977*, 2021. (Cited on pages 1, 2, 7, 9, and 20.)

[35] Yifan Hu, Xin Chen, and Niao He. Sample complexity of sample average approximation for conditional stochastic optimization. *SIAM Journal on Optimization*, 30(3):2103–2133, 2020. (Cited on page 34.)

[36] Yifan Hu, Xin Chen, and Niao He. On the bias-variance-cost tradeoff of stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 34, pages 22119–22131, 2021. (Cited on page 7.)

[37] Yifan Hu, Jie Wang, Yao Xie, Andreas Krause, and Daniel Kuhn. Contextual stochastic bilevel optimization. *Advances in Neural Information Processing Systems*, 36, 2024. (Cited on pages 2, 3, 7, 19, and 27.)

[38] Yifan Hu, Siqi Zhang, Xin Chen, and Niao He. Biased stochastic first-order methods for conditional stochastic optimization and applications in meta learning. *Advances in Neural Information Processing Systems*, 33:2759–2770, 2020. (Cited on page 34.)

[39] Yujing Hu, Weixun Wang, Hangtian Jia, Yixiang Wang, Yingfeng Chen, Jianye Hao, Feng Wu, and Changjie Fan. Learning to utilize shaping rewards: A new approach of reward shaping. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15931–15941. Curran Associates, Inc., 2020. (Cited on page 19.)

[40] Jiawei Huang, Vinzenz Thoma, Zebang Shen, Heinrich H. Nax, and Niao He. Learning to steer markovian agents under model uncertainty, 2024. (Cited on page 19.)

[41] Dima Ivanov, Paul Dütting, Inbal Talgam-Cohen, Tonghan Wang, and David C. Parkes. Principal-agent reinforcement learning, 2024. (Cited on pages 2, 20, and 21.)

[42] Sham Kakade. A natural policy gradient. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, page 1531–1538, Cambridge, MA, USA, 2001. MIT Press. (Cited on page 52.)

[43] Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. *Advances in neural information processing systems*, 34:30271–30283, 2021. (Cited on page 19.)

[44] Jeongyeol Kwon, Dohyun Kwon, and Hanbaek Lyu. On the complexity of first-order methods in stochastic bilevel optimization. *arXiv preprint arXiv:2402.07101*, 2024. (Cited on page 19.)

[45] Jeongyeol Kwon, Dohyun Kwon, Stephen Wright, and Robert D Nowak. A fully first-order method for stochastic bilevel optimization. In *International Conference on Machine Learning*, pages 18083–18113. PMLR, 2023. (Cited on page 19.)

[46] Joshua Letchford and Yevgeniy Vorobeychik. Optimal interdiction of attack plans. In *AAMAS*, pages 199–206. Citeseer, 2013. (Cited on page 2.)

[47] Boyi Liu, Jiayang Li, Zhuoran Yang, Hoi-To Wai, Mingyi Hong, Yu Nie, and Zhaoran Wang. Inducing equilibria via incentives: Simultaneous design-and-play ensures global convergence. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 29001–29013. Curran Associates, Inc., 2022. (Cited on pages 4 and 19.)

[48] Risheng Liu, Xuan Liu, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. A value-function-based interior-point method for non-convex bi-level optimization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6882–6892. PMLR, 18–24 Jul 2021. (Cited on page 19.)

[49] Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6820–6829. PMLR, 13–18 Jul 2020. (Cited on pages 19, 43, 48, 51, and 52.)

[50] Alberto Maria Metelli, Mirco Mutti, and Marcello Restelli. Configurable markov decision processes. In *International Conference on Machine Learning*, pages 3491–3500. PMLR, 2018. (Cited on pages 2, 3, and 19.)

[51] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 2772–2782, Red Hook, NY, USA, 2017. Curran Associates Inc. (Cited on pages 4, 48, and 50.)

[52] ATD Perera and Parameswaran Kamalaruban. Applications of reinforcement learning in energy systems. *Renewable and Sustainable Energy Reviews*, 137:110618, 2021. (Cited on page 1.)

[53] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014. (Cited on page 1.)

[54] Guannan Qu and Adam Wierman. Finite-time analysis of asynchronous stochastic approximation and q-learning. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 3185–3205. PMLR, 09–12 Jul 2020. (Cited on pages 49 and 50.)

[55] Giorgia Ramponi, Alberto Maria Metelli, Alessandro Concetti, and Marcello Restelli. Learning in non-cooperative configurable markov decision processes. *Advances in Neural Information Processing Systems*, 34:22808–22821, 2021. (Cited on pages 2 and 19.)

[56] R. Tyrrell Rockafellar and Roger J. B. Wets. *Variational Analysis*. Springer Berlin Heidelberg, 1998. (Cited on page 46.)

[57] Aaron Roth, Jonathan Ullman, and Zhiwei Steven Wu. Watch and learn: Optimizing from revealed preferences feedback. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 949–962, 2016. (Cited on page 10.)

[58] Han Shen, Zhuoran Yang, and Tianyi Chen. Principled penalty-based methods for bilevel reinforcement learning and rlhf. *arXiv preprint arXiv:2402.06886*, 2024. (Cited on pages 2, 3, 6, 19, and 20.)

[59] Laixi Shi, Eric Mazumdar, Yuejie Chi, and Adam Wierman. Sample-efficient robust multi-agent reinforcement learning in the face of environmental uncertainty. *arXiv preprint arXiv:2404.18909*, 2024. (Cited on page 19.)

[60] Arunesh Sinha, Fei Fang, Bo An, Christopher Kiekintveld, and Milind Tambe. Stackelberg security games: looking beyond a decade of success. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 5494–5501, 2018. (Cited on page 2.)

[61] Heinrich von Stackelberg. *Marktform und Gleichgewicht*. Klassiker der Nationalökonomie. Verlag Wirtschaft und Finanzen, Düsseldorf, 1934. (Cited on page 19.)

[62] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. (Cited on page 1.)

[63] Matteo Turchetta, Andrey Kolobov, Shital Shah, Andreas Krause, and Alekh Agarwal. Safe reinforcement learning via curriculum induction. *Advances in Neural Information Processing Systems*, 33:12151–12162, 2020. (Cited on page 19.)

[64] Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos, and Matthieu Geist. Leverage the average: an analysis of kl regularization in reinforcement learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. (Cited on page 21.)

[65] Jing Wang, Meichen Song, Feng Gao, Boyi Liu, Zhaoran Wang, and Yi Wu. Differentiable arbitrating in zero-sum markov games. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, page 1034–1043, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems. (Cited on pages 2 and 19.)

[66] Kai Wang, Lily Xu, Andrew Perrault, Michael K. Reiter, and Milind Tambe. Coordinating followers to reach better equilibria: End-to-end gradient descent for stackelberg games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5):5219–5227, Jun. 2022. (Cited on page 19.)

[67] Xu Wang, Sen Wang, Xingxing Liang, Dawei Zhao, Jincai Huang, Xin Xu, Bin Dai, and Qiguang Miao. Deep reinforcement learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. (Cited on page 1.)

[68] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992. (Cited on page 51.)

[69] Jibang Wu, Siyu Chen, Mengdi Wang, Huazheng Wang, and Haifeng Xu. Contractual reinforcement learning: Pulling arms with invisible hands, 2024. (Cited on pages 2, 20, and 21.)

[70] Shuo Wu, Haoxiang Ma, Jie Fu, and Shuo Han. Robust reward design for markov decision processes, 2024. (Cited on page 19.)

[71] Li Xia, Luyao Zhang, and Peter W. Glynn. Risk-sensitive markov decision processes with long-run cvar criterion. *Production and Operations Management*, 32(12):4049–4067, December 2023. (Cited on page 19.)

[72] Zeke Xie, Issei Sato, and Masashi Sugiyama. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *International Conference on Learning Representations*, 2021. (Cited on page 9.)

[73] Chang Yang, Yuiyu Wang, Xinrun Wang, and Zhen Wang. A game-theoretic perspective of generalization in reinforcement learning. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022. (Cited on page 19.)

[74] Jiachen Yang, Ang Li, Mehrdad Farajtabar, Peter Sunehag, Edward Hughes, and Hongyuan Zha. Learning to incentivize other learning agents. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15208–15219. Curran Associates, Inc., 2020. (Cited on page 19.)

[75] Yan Yang, Bin Gao, and Ya xiang Yuan. Bilevel reinforcement learning via the development of hyper-gradient without lower-level convexity, 2024. (Cited on pages 3 and 19.)

[76] Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2021. (Cited on page 1.)

[77] Guanghui Yu and Chien-Ju Ho. Environment design for biased decision makers. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 592–598. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track. (Cited on page 20.)

[78] Haifeng Zhang, Jun Wang, Zhiming Zhou, Weinan Zhang, Ying Wen, Yong Yu, and Wenxin Li. Learning to design games: strategic environments in reinforcement learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 3068–3074. AAAI Press, 2018. (Cited on pages 2, 5, and 19.)

[79] Haoqi Zhang and David Parkes. Value-based policy teaching with active indirect elicitation. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 1*, AAAI'08, page 208–214. AAAI Press, 2008. (Cited on page 20.)

[80] Kaiqing Zhang, Alec Koppel, Hao Zhu, and Tamer Başar. Global convergence of policy gradient methods to (almost) locally optimal policies. *SIAM Journal on Control and Optimization*, 58(6):3586–3612, January 2020. (Cited on page 46.)

[81] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pages 321–384, 2021. (Cited on page 19.)

[82] Stephan Zheng, Alexander Trott, Sunil Srinivasa, David C. Parkes, and Richard Socher. The ai economist: Taxation policy design via two-level deep multiagent reinforcement learning. *Science Advances*, 8(18):eabk2607, 2022. (Cited on pages 2, 19, and 20.)

Contents

A Related Work

Stochastic bilevel optimization has been extensively explored in the literature [21, 5]. In recent years, there is a pivotal shift to non-asymptotic analysis of stochastic gradient methods [28, 16, 43, 45, 44, 48]. [37] propose contextual stochastic bilevel optimization where the lower level solves a static contextual optimization. Our work generalizes to the lower level solving a contextual MDP. This poses unique challenges in terms of hypergradient estimation and sample generation. Comparing to bilevel optimization, leveraging the special structure of CB-RL, we avoid Hessian and Jacobian estimation of the lower-level MDP when computing the hyper policy gradient, which is crucial for scalability.

Configurable MDP [50] is an extension of a traditional MDP allowing external parameters or settings to be adjusted by the decision-maker, often referred to as the *configurator*. Only recently some works studied the case where the configurator has a different objective than the agent [55]. However, that work assumes access to a finite number of parameters that the configurator can control, while our model goes beyond this assumption. In addition, our model captures the variability and uncertainty that the agent could face in the same configuration environment.

Stackelberg games are a game theoretic framework, where a leader takes actions to which one or multiple followers choose the best response [61]. Several existing lines of work have studied solving variants of Stackelberg games. Examples include Stackelberg equilibrium solvers [24, 27], opponent shaping [25, 74], mathematical programs with equilibrium constraints [47, 65, 66], inducing cooperation [6, 4], steering economic simulations [19, 82]. These works are either too general with limited implications for our problem or consider entirely distinct settings.

Multi-agent RL (MARL) studies multiple agents interacting in a joint environment, i.e., their actions together determine the next state [81, 59]. In CB-RL the lower level CMDPs can be seen as a special instance of MARL where the interactions of the followers are restricted to jointly influencing the decision of the leader.

Bilevel RL studies how to design additional rewards or change the underlying MDP to achieve desirable learning outcomes. Many applications are formulated as bilevel RL, such as environment design for generalization [22, 23, 73], reward shaping [31, 39, 40, 70], safe reinforcement learning [63], optmizing conditional value at risk[71], and model design [15, 78, 11]. Previous work on bilevel RL considers a special case of our setting when there is only one lower-level MDP [15, 13, 58, 75]. In particular, [15] focus on the case when the leader has control on the follower's training procedure. [58] further extend from one single lower-level MDP to a lower-level min-max game. [13] and the concurrent work of [75] focus on the case when the leader can only influence the reward of the MDP.

The introduction of the context makes CB-RL harder to solve as there can be many followers, each with its own preferences, and their best response policies change even for the same leader decision x when facing different contextual uncertainties. Multiple followers and additional side information are very common, which highlights the practical relevance of our work. In addition, the algorithms in the aforementioned works focus on deterministic updates on the upper and lower level decisions, i.e., assuming access to the full hypergradient and performing exact policy gradient/value iteration, which is both computationally hard and not feasible for large-scale practical applications. To the best of our knowledge, we are the first to provide a convergence analysis for the stochastic case, when the hypergradient is estimated from samples and the lower level uses a stochastic update rule.

B Applications: RLHF, Tax Design, Reward Shaping, Contract Design, and Dynamic Mechanism Design

We list several applications of CB-RL below. For a clearer exposition, we omit the entropy regularization term at the lower level. However, we stress that some of the referenced works explicit use entropy regularization [18, 15] or make overlapping assumptions such as unique optimal policies [13]. Additionally, for problems without explicit entropy regularization we refer to [15, 49, 20, 26] who have shown that entropy-regularized RL approximates the unregularized problem as $\lambda \to 0$ both in the upper and lower level.

Reinforcement Learning from human feedback (RLHF) considers the setting where an agent tries to learn a task from human feedback. The difficulty stems from the fact that the human feedback is

given as preferences over two possible trajectories and not directly as a reward [17]. The feedback is of the form $\{\tau_0, \tau_1, l\}$ where τ_0, τ_1 are two trajectories and $l \in 0, 1$ indicates whether τ_0 is preferred over τ_1 or vice versa. It has been shown that this problem can be framed as CB-RL, where the upper-level tries to learn rewards that minimize the cross-entropy loss, between the actual and predicted labels, using the Bradley-Terry Model and the lower level finds the optimal policy with respect to that reward function [13, 58],

$$\max_x \mathbb{E}_{\tau_0,\tau_1\sim\mathcal{D}(\pi^*_x),l}\left[(1-l)\log\mathbb{P}\left(\tau_0 \succ \tau_1|r_x\right) + l\log\mathbb{P}\left(\tau_1 \succ \tau_0|r_x\right)\right]$$
$$\text{s.t. } \pi^*_x(\cdot) = \operatorname*{argmax}_\pi \mathbb{E}\left[\sum_{t=0}^{H}\gamma^t r_x(s_t, a_t)\right].$$

Here H is the time horizon. D is the sampling distribution of trajectories using $\pi^*_{x,\xi}$ and

$$\mathbb{P}\left(\tau_0 \succ \tau_1|r_x\right) = \frac{\exp\sum_{t=0}^{H}\gamma^t r_x(s_t^0, a_t^0)}{\exp\sum_{t=0}^{H}\gamma^t r_x(s_t^0, a_t^0) + \exp\sum_{t=0}^{H}\gamma^t r_x(s_t^1, a_t^1)}.$$

Note in the case of standard RLHF the context becomes trivial.

Tax Design for Macroeconomic Modeling considers a public entity setting tax rates and representative households responding optimally by balancing their short-term utility of consumption and long-term wealth accumulation [34, 15, 82]. A potential formulation of this problem as CB-RL is

$$\max_{x,y}\mathbb{E}_\xi\left[\phi(x, y, \pi^*_{x,y,\xi}, \xi)\right] \text{ s.t. } \pi^*_{x,y,\xi}(\cdot) = \operatorname*{argmax}_\pi \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t\left(r^W_\xi(s_t) + r^C_{y,\xi}(\pi(s_t))\right)\right],$$

where x and y denote the income and value-added tax rates, respectively, and ϕ defines the social welfare objective of the leader. The state s_t defines the wealth of a household while their actions decide their working hours and consumption in each time step. The reward function r^W_ξ and $r^C_{y,\xi}$ define the households' utility functions for wealth and consumption, respectively. The value-added tax rate y affects the consumption utility function $r^C_{y,\xi}$ while the income tax x changes the transition kernel modeling wealth accumulation, i.e., $s_{t+1} \sim P_{x,\xi}(\cdot; s_t, a_t)$. ξ represents the preferences of the households over several consumption goods and their productivity in this problem formulation.

Population Principal-Agent Reward Shaping considers a principal aiming to craft a non-negative bonus reward function r^B_x, parameterized by x, to motivate an agent [8, 77, 79]. Commonly, a principal faces multiple agents that form a distribution. Each agent has its own individual reward function r_ξ. This scenario, termed *population principal-agent reward shaping* is captured by our CB-RL framework.

$$\max_x \mathbb{E}_\xi^{\pi^*_{x,\xi}}\left[\sum_{t=0}^{\infty}\gamma^t\overline{r}(s_t, \pi^*_{x,\xi}(s_t))\right] \text{ s.t. } \pi^*_{x,\xi}(\cdot) = \operatorname*{argmax}_\pi \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t\left(r_\xi(s_t, \pi(s_t)) + r^B_x(s_t, \pi(s_t))\right)\right].$$

Here $\mathbb{E}_\xi$ denotes the expectation over the distribution of agents and the trajectories. The policy $\pi^*_{x,\xi}(\cdot)$ is the optimal response of the ξ-th agent to the composite reward function $r_\xi + r^B_x$. The principal's reward is $\overline{r}(s_t, a_t)$ when the agent visits the state action pair (s_t, a_t).

Dynamic Contract Design studied by [41, 69] is similar to the above reward shaping. Generalizing it to a contextual setting, the problem consists of an agent of type ξ that incurs a cost $c_\xi(s, a)$ for taking action a in state s. The principal in turn gets a reward $r(s, s')$ for transitioning from state s to s', but cannot observe the agent's action. It can however offer contracts $x(s, s')$ that get paid if the MDP transitions from state s to s'. These contracts are positive payments by the principal and are thus added to the lower-level objective and substracted from the upper-level objective.

$$\max_x \mathbb{E}_{\xi,\pi^*_{x,\xi}}\left[\sum_{t=0}^{\infty}\gamma^t\left(r(s_t, s_{t+1}) - x(s_t, s_{t+1})\right)\right]$$
$$\text{s.t. } \pi^*_{x,\xi}(\cdot) = \operatorname*{argmax}_\pi \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t\left(x(s_t, s_{t+1}) - c_\xi(s_t, a_t)\right)\right].$$

Dynamic Mechanism Design considers the problem of a mechanism designer controlling an MDP for a group of n bidders, who get a reward based on the observed trajectories [18]. The context ξ parameterizes the bidders' reward functions $r_{i,\xi}$, which they report to the mechanism designer. The latter wants to learn a policy for the MDP and charge payments to the bidders, to ensure eliciting truthful reward reports and also maximizie an objective $\mathcal{L}$, e.g. the total sum of payments. In this setting, [18] propose to search for such a mechanism within the class of affine maximizers, as they guarantee truthful reports by all bidders. In these mechanisms, a set of agent-dependent weights $x_{w,i}$ and state-action dependent boosts x_b is chosen by the mechanism designer, then a policy π is learned to maximize the corresponding affinely transformed social welfare $\mathbb{E}_{s_t,a_t\sim\pi}\left[\sum_{t=0}^{T}\left(\sum_{i=1}^{n} x_{w,i} r_{i,\xi}(s_t,a_t)\right) + x_b(s_t,a_t)\right]$ and bidders are charged for the learned policy depending on their reported reward functions. Searching for the optimal mechanism parameters $x_{w,i}$ and x_b to maximize $\mathcal{L}$ in expectation over ξ, subject to the constraint that the mechanism's policy maximizes affine social welfare can be formulated as CB-RL. In this case $x_{w,i}$ and x_b are the decision variable, the context parameterizes the bidders' preferences and the affinely transformed social welfare at each time step is the reward function of the lower-level MDP, as shown below:

$$\min_{x_w,x_b} \mathbb{E}_\xi[\mathcal{L}\left(\pi^*_{\xi,x_w,x_b}, x_w, x_b\right)]$$
$$\text{s.t. } \pi^*_{\xi,x_w,x_b} = \arg\max_\pi \mathbb{E}_{s_t,a_t\sim\pi}\left[\sum_{t=0}^{T}\left(\sum_{i=1}^{n} x_{w,i} r_{i,\xi}(s_t,a_t)\right) + x_b(s_t,a_t)\right].$$

Note, that all previous works in these application areas have either focused on the setting with a single representative follower [8, 15, 41, 69] or presented a problem-specific algorithm that cannot capture our CB-RL framework in its full generality [8, 18].

The CB-RL framework is also related to **Meta reinforcement learning (Meta RL)**, that aims to leverage the similarity of several RL tasks to learn common knowledge and use it on new unseen tasks [7]. One way to formulate Meta RL problems is to find a common regularization policy $\tilde{\pi}$ for multiple tasks.

$$\max_{\tilde{\pi}} \mathbb{E}_\xi\left[\sum_{t=0}^{\infty}\gamma^t r_\xi(s_t,\pi^*_{\tilde{\pi},\xi}(s_t))\right] \text{ s.t. } \pi^*_{\tilde{\pi},\xi}(\cdot) = \operatorname*{argmax}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r_\xi(s_t,\pi(s_t)) - \frac{\lambda}{2}KL(\pi(s_t)||\tilde{\pi}(s_t))\right],$$

where ξ represents the distribution of multiple RL tasks and r_ξ is the reward for the task indexed by ξ. Although the upper-level regularizer is different from entropy-regularization it still results in a unique softmax policy of the form $\pi^*_{s_\xi}(s,a) \propto \exp\left(\frac{Q_{s_\xi,\pi}(s,a)}{\lambda} + \log(\tilde{\pi}(s,a))\right)$ [64]. Moreover, in Meta RL the leader does not change the transitions or rewards, but the target policy $\tilde{\pi}$, that goes into the KL regularization term of the followers.

C Frequently-Used Notation

Table 4: Table of notation used in the paper.

Notation	Description
x	Upper-level decision variable/leader's decision
ξ	Contextual variable
$\mathcal{M}_{x,\xi}$	MDP parameterized by x and ξ
$\mathcal{S}$	State Space
$\mathcal{A}$	Action Space
$r_{x,\xi}$	Reward function
$P_{x,\xi}$	Transition kernel
$\mu_{x,\xi}$	Initial state distribution
γ	Discount factor
τ	Trajectories of MDP
$\pi_{x,\xi}$	Follower policy for $\mathcal{M}_{x,\xi}$
$J_{\lambda,x,\xi}(\pi)$	Entropy-regularized lower-level objective function
s_0	Initial state
$H(\pi; s)$	Entropy of policy π at state s
$V^{\pi}_{\lambda,x,\xi}(s)$	Value function
$Q^{\pi}_{\lambda,x,\xi}(s,a)$	Q-function
$A^{\pi}_{\lambda,x,\xi}(s,a)$	Advantage function
$\pi^{*}_{x,\xi}$	Optimal policy maximizing $J_{\lambda,x,\xi}(\pi)$ (dependence on λ is droped)
$\pi^{o}_{x,\xi}$	Oracle policy with distance δ from $\pi^{*}_{x,\xi}$
$\pi^{t_k}_{x,\xi}$	Follower policy after performing t_k learning steps
$f(x, \pi_{x,\xi}, \xi)$	Upper-level loss for specific context ξ
$F(x)$	Upper-level loss function
λ	Regularization parameter
L_f	Lipschitz continuity parameter of f
S_f	Smoothness parameter of f
$\overline{R}$	Upper bound on absolute value of reward function
K_1, K_2	$\left\|\partial_x \log P_{x,\xi}(s'; s, a)\right\|_{\infty} < K_1$, $\left\|\partial_x r_{x,\xi}(s,a)\right\|_{\infty} < K_2$.
δ	Oracle inaccuracy, such that $\forall x, \forall \xi : \mathbb{E}_o \left[\left\| \pi^{*}_{x,\xi} - \pi^{o}_{x,\xi} \right\|^2_{\infty} \right] \leq \delta^2$
RT-Q	Randomnly-Truncated Soft Q-learning
CB-RL	Contextual Bilevel Reinforcement Learning
HPGD	Hyper Policy Gradient Descent
K	Used to define bias,variance and complexity of RT-Qneu
ν	Sampling distribution to estimate hypergradient
m	$m := \min_s \nu(s)$
a	test
$\widehat{\partial_x A^{\pi^{t_k}}}(s,a)$	Estimate of Advantage derivative, obtained from Algorithm 2
$\widehat{\partial_x Q^{\pi^{t_k}}}(s,a)$	Estimate of Q derivative, obtained from Algorithm 2.
$\widetilde{\partial_x Q^{\pi^{t_k}}}$	Smoothed Q derivative setimate, $\widetilde{\partial_x Q^{\pi^{t_k}}} := \frac{1}{2^k} \sum_{l=1}^{2^k} \widehat{\partial_x Q^{\pi^{t_k}}}(\tau_l)$
$\overline{r}$	Upper-Level reward function for special loss function (cf. Section 3.3)
$\overline{V}, \overline{Q}$	Unregularized upper-level value and Q functions for $\overline{r}$
Geo$(1-\gamma)$	Geometric distribution with parameter $1-\gamma$
$\mathcal{T}^{*}_{\lambda}$	Soft Bellman optimality operator

D Algorithms

We give the pseudocode to certain algorithms/routines/procedures mentioned in the main text.

Algorithm 2 `GradientEstimator`(ξ, x, s, a, o)

Input: ξ, x state s, action a, trajectory oracle o
$T_Q, T_V \sim \text{Geo}(1-\gamma), T'_Q, T'_V \sim \text{Geo}(1-\gamma^{0.5})$
$\tau_Q \leftarrow \texttt{SampleTrajectory}(o, \texttt{start} = (s,a), \texttt{length} = T_Q + T'_Q + 1)$
$\tau_V \leftarrow \texttt{SampleTrajectory}(o, \texttt{start} = s, \texttt{length} = T_V + T'_V + 1)$
$\widehat{\frac{d}{dx}Q}(s,a) \leftarrow \sum_{t=0}^{T_Q} \frac{d}{dx} r(s_t^{\tau_Q}, a_t^{\tau_Q}) +$
$\frac{\gamma}{1-\gamma} \frac{d}{dx} \log P(s_{T_Q+1}^{\tau_Q}; s_{T_Q}^{\tau_Q}, a_{T_Q}^{\tau_Q}) \sum_{t=T_Q+1}^{T_Q+T'_Q+1} \gamma^{(t-T_Q-1)/2} \left(r(s_t^{\tau_Q}, a_t^{\tau_Q}) + \lambda H(\pi(\cdot; s_t)) \right)$
$\widehat{\partial_x V}(s) \leftarrow \sum_{t=0}^{T_V} \partial_x r(s_t^{\tau_V}, a_t^{\tau_V}) +$
$\frac{\gamma}{1-\gamma} \partial_x \log P(s_{T_V+1}^{\tau_V}; s_{T_V}^{\tau_V}, a_{T_V}^{\tau_V}) \sum_{t=T_V+1}^{T_V+T'_V+1} \gamma^{(t-T_V-1)/2} \left(r(s_t^{\tau_V}, a_t^{\tau_V}) + \lambda H(\pi(\cdot; s_t)) \right)$
Output: $\widehat{\partial_x A(s,a)} \leftarrow \widehat{\partial_x Q}(s,a) - \widehat{\partial_x V}(s)$

Algorithm 3 Soft Value Iteration

Input: Number of iterations T
Result: Approximation $V_\lambda \approx V_\lambda^*$, policy $\pi_\lambda \approx \pi_\lambda^*$
Initialize $V_\lambda = 0$
for $t = 0$ **to** T **do**
  for $s \in \mathcal{S}$ **do**
    for $a \in \mathcal{A}$ **do**
      $Q_\lambda(s,a) = r(s,a) + \gamma \mathbb{E}_{s'|s,a} \left[V_\lambda(s') \right]$
    end for
    $V_{\text{new},\lambda}(s) = \lambda \log \left(\sum_{a \in \mathcal{A}} \exp \left(\frac{Q_\lambda(s,a)}{\lambda} \right) \right)$
  end for
  set $V_\lambda := V_{\text{new},\lambda}$
end for
$\pi_\lambda^o \leftarrow \frac{\exp(Q_\lambda(s,a)/\lambda)}{\sum_a \exp(Q_\lambda(s,a)/\lambda)}$
return V_λ and π_λ^o

Algorithm 4 `SoftQlearning`$(T, \pi_B, \{\alpha_t\}_{t \geq 0})$

Input: Number of iterations T, Behavioural Policy π_B, Stepsizes $\{\alpha_t\}_{t \geq 0}$
Result: Approximation $Q_\lambda \approx Q_\lambda^*$, policy $\pi_\lambda \approx \pi_\lambda^*$
Initialize $Q_\lambda = 0$
Initialise s_0
for $t = 0$ **to** T **do**
  Sample $a \sim \pi_B(\cdot; s_t)$
  Observe next reward $r(s_t, a)$ and state $s_{t+1} \sim P(\cdot|s_t, a)$
  $Q_\lambda(s_t,a) = Q_\lambda(s_t,a) + \alpha_t \left(r(s_t,a) + \gamma\lambda \log \left(\sum_{a' \in \mathcal{A}} \exp \left(\frac{Q_\lambda(s_{t+1},a')}{\lambda} \right) \right) \right)$
end for
$\pi_\lambda^o(a; s) \leftarrow \frac{\exp(Q_\lambda(a|s)/\lambda)}{\sum_{a'} \exp(Q_\lambda(s,a')/\lambda)}$
return Q_λ and π_λ^o

Algorithm 5 `DecomposableGradientEstimator`

Input: ξ, x, initial distribution $\mu_{x,\xi}$, oracle o
$T, \sim \text{Geo}(1-\gamma)$, $T' \sim \text{Geo}(1-\gamma^{0.5})$
$(s_0, a_0, \ldots, s_{T+T'}, a_{T+T'}) \leftarrow \texttt{SampleTrajectory}(o, \texttt{start} = \mu_{x,\xi}, \texttt{length} = T + T')$
$\widehat{A^{\pi^o_{x,\xi}}_{\lambda,x,\xi}}(s_T, a_T) \leftarrow \texttt{GradientEstimator}(\xi, x, s_T, a_T, o)$
$\widehat{\frac{dF}{dx}} = \left(\sum_{t=0}^{T} \frac{d}{dx}\overline{r}(s_t, a_t)\right) + \frac{1}{\lambda(1-\gamma)}\widehat{\partial_x A^{\pi^o_{x,\xi}}_{\lambda,x,\xi}}(s_T, a_T) \sum_{t'=T}^{T+T'} \gamma^{(t-T)/2}\overline{r}(s_{t'}, a_{t'})$
$+\frac{1}{1-\gamma}\partial_x \log P(s_T, a_{T-1}, s_{T-1}) \sum_{t'=T}^{T+T'} \gamma^{(t'-T)/2}\overline{r}(s_{t'}, a_{t'})$
Output: $\widehat{\frac{dF}{dx}}$

Algorithm 6 Vanilla Policy Gradient Algorithm

Data: Initial parameter θ_0, initial state s
Result: Approximate policy π_{θ_L}
for $l = 0$ **to** L **do**
Sample $T \sim \text{Geo}(1-\gamma)$
Sample trajectory $(s_0, a_0, s_1, \ldots, a_{T-1}, s_T, r_T, a_T)$ using policy π_{θ_l}
Sample $T' \sim \text{Geo}(1-\gamma^2)$
Set $\tilde{s}_0 = s_{T'}$ and $\tilde{a}_0 = a_T$
Sample trajectory $(\tilde{s}_0, \tilde{a}_0, \tilde{s}_1, \ldots, \tilde{a}_{T'-1}, \tilde{s}_{T'}, \tilde{r}_{T'}, \tilde{a}_{T'})$ using policy π_{θ_l}
Determine step-size α.
$\widehat{\nabla J}_s(\theta_l) = \frac{1}{1-\gamma}\nabla \log \pi_{\theta_l}(a_T|s_T) \sum_{t'=0}^{T'-1} \gamma^{t'/2}\tilde{r}_{t'+1}$
$\theta_{l+1} = \theta_l - \alpha\widehat{\nabla J}_s(\theta_l)$
end for

E Proofs

E.1 Overview

In this section, we provide proofs for the presented theorems and propositions. We provide the proof of Theorem 1, deriving the hypergradient of function $F(x)$; the proof of Theorem 2, deriving the derivative of the action-value function with respect to x; the proof of our main result, Theorem 3, which shows convergence of HPGD to a stationary point of $F(x)$.

For the propositions, we show how to estimate the upper-level gradient if f is decomposable in the proof of Proposition 1; In Proposition 2 we show how to compute the gradient of the optimal policy with respect to x; the proof of Proposition 3, which shows we can achieve unbiased estimates of the advantage hypergradient; and the proof of Proposition 4, which shows the same for the special case when f decomposes.

We state and proof Propositions 5 to 8 which show convergence in L_2 to the optimal policy of soft value iteration, soft Q-learning, Vanilla Policy Gradient and Natural Policy Gradient respectively.

Lastly, we prove Theorem 4, regarding the reduced iteration complexity of RT-Q claimed in Section 4.

E.2 Proof of Theorem 1

Proof. The proof relies on on three main ideas.

1. Show that Dominated Convergence applies, i.e. all derivatives are uniformly bounded by an integrable function. Thus we can exchange derivative and expectation and compute the derivative of f instead of F.

2. Use Proposition 2 to get an expression for the derivative of the optimal policy with respect to x.

3. Use importance sampling with some distribution ν to get an expression of the hypergradient that we can cheaply sample, instead of having to multiply two matrices with size $|\mathcal{S}| \times |\mathcal{A}|$.

By Proposition 2, it follows that

$$\begin{aligned}\left\| \frac{\partial \pi^*_{x,\xi}}{\partial x} \right\|_\infty &\leq \frac{2}{\lambda} \left\| \partial_x Q^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a) \right\|_\infty \\ &\leq \frac{2}{\lambda} \frac{K_2}{1-\gamma} \frac{K_1 \overline{R} K_1}{(1-\gamma)^2}.\end{aligned}$$

As the partial derivatives of f are bounded by L_f (cf. Assumption 3.1), we can apply the Dominated Convergence Theorem to get

$$\begin{aligned}\partial_x \mathbb{E}\left[f(x, \pi^*_{x,\xi}, \xi)\right] &= \mathbb{E}\left[\partial_x f(x, \pi^*_{x,\xi}, \xi)\right] \\ &= \mathbb{E}\left[\frac{\partial_1 f(x, \pi^*_{x,\xi}, \xi)}{\partial x} + \frac{\partial_2 f(x, \pi^*_{x,\xi}, \xi)}{\partial \pi^*_{x,\xi}} \frac{\partial \pi^*_{x,\xi}}{\partial x}\right] \\ &= \mathbb{E}\left[\frac{\partial_1 f(x, \pi^*_{x,\xi}, \xi)}{\partial x} + \sum_{s,a} \frac{\partial_2 f(x, \pi^*_{x,\xi}, \xi)}{\partial \pi^*_{x,\xi}(a;s)} \frac{\partial \pi^*_{x,\xi}(a;s)}{\partial x}\right] \\ &= \mathbb{E}\left[\frac{\partial_1 f(x, \pi^*_{x,\xi}, \xi)}{\partial x} + \sum_{s,a} \frac{\partial_2 f(x, \pi^*_{x,\xi}, \xi)}{\partial \pi^*_{x,\xi}(a;s)} \frac{1}{\lambda} \pi^*_{x,\xi}(a;s) \partial_x A^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a)\right] \qquad (9) \\ &= \mathbb{E}\left[\frac{\partial_1 f(x, \pi^*_{x,\xi}, \xi)}{\partial x} + \mathbb{E}_{s\sim\nu, a\sim\pi^*_{x,\xi}}\left[\frac{1}{\lambda\nu(s)} \frac{\partial_2 f(x, \pi^*_{x,\xi}, \xi)}{\partial \pi^*_{x,\xi}(a;s)} \partial_x A^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a)\right]\right],\end{aligned}$$

where we use Proposition 2 for eq. (9) and importance sampling with any distribution ν in the last equality, as long as ν has full support on $\mathcal{S}$. Further, we note that $\frac{\partial_2 f(x, \pi^*_{x,\xi}, \xi)}{\partial \pi^*_{x,\xi}} \in \mathrm{Mat}_{1,|S|\times|A|}(\mathbb{R})$ and $\frac{\partial \pi^*_{x,\xi}}{\partial x} \in \mathrm{Mat}_{|S|\times|A|,d}(\mathbb{R})$. Hence, we just explicitely write out the matrix multiplication for the second equality. The second equality follows from the multivariate chain rule and the first equality from the Dominated Convergence Theorem. □

E.3 Proof of Theorem 2

Proof. Theorem 2 is important as $\partial_x A^\pi_{\lambda,x,\xi}(s,a)$ and thus $\partial_x Q^\pi_{\lambda,x,\xi}(s,a)$ allow us to compute $\frac{d\pi^*_{x,\xi}}{dx}$ (cf. Proposition 2). We will prove the theorem using an induction proof, which follows the analysis of the policy gradient theorem. However, instead of at each timestep $\pi(a_t; s_t)$ depending on a policy parameter θ, it will be the transition $P_{x,\xi}(s_{t+1}; s_t, a_t)$ and reward $r_{x,\xi}(s_t, a_t)$ depending on x. Also note that we only consider the partial derivative with respect to x, evaluated at some policy π, which in the case of $\pi^*_{x,\xi}$

In the following, we will show by induction that

$$\frac{dQ^\pi_{\lambda,x,\xi}(s,a)}{dx} = \sum_{t=0}^{\infty} \sum_{s',a'} \gamma^t p_{x,\xi}(s, a \to s', a'; t, \pi) \left(\frac{dr_{x,\xi}(s',a')}{dx} + \gamma \sum_{s''} \frac{dP_{x,\xi}(s''; s', a')}{dx} V^\pi_{\lambda,x,\xi}(s'') \right),$$

where $p_{x,\xi}(s, a \to s', a'; t, \pi)$ is the probability that the Markov Chain induced by π, starting from s, a reaches s', a' after t steps. Note, the formulation is equivalent to the one stated in Theorem 2.

The proof follows the analysis of the standard policy gradient theorem. We drop here the dependence on x and ξ to simplify the notation. Assuming that $Q^\pi_\lambda(s,a)$ is differentiable for all s, a, we show by induction that for all $n \in \mathbb{N}$ it holds that

$$\begin{aligned}\frac{dQ^\pi_\lambda(s,a)}{dx} =& \sum_{t=0}^{n} \sum_{s',a'} \gamma^t p(s, a \to s', a'; t, \pi) \left(\frac{dr(s',a')}{dx} + \gamma \sum_{s''} \frac{dP(s''; s', a')}{dx} V^\pi_\lambda(s'') \right) \\ &+ \gamma^{n+1} \sum_{\tilde{s},\tilde{a}} p(s, a \to \tilde{s}, \tilde{a}; n+1, \pi) \frac{dQ^\pi_\lambda(\tilde{s}, \tilde{a})}{dx}.\end{aligned} \qquad (10)$$

The claim then follows as $n \to \infty$.

Base case $(n = 0)$ It is easy to check that

$$\begin{aligned}
\frac{dQ^\pi_\lambda(s,a)}{dx} =& \frac{d}{dx}\left(r(s,a) + \gamma \sum_{s'} P(s';s,a)V_\lambda(s')\right) \\
=& \frac{d}{dx}r(s,a) + \gamma \sum_{s'}\left(\frac{d}{dx}P(s';s,a)V^\pi_\lambda(s') + P(s';s,a)\frac{d}{dx}V^\pi_\lambda(s')\right) \\
=& \frac{d}{dx}r(s,a) + \gamma \sum_{s'}\left(\frac{d}{dx}P(s';s,a)V^\pi_\lambda(s') + P(s';s,a)\sum_{a'}\pi(a';s')\frac{d}{dx}Q^\pi_\lambda(s',a')\right) \\
=& \sum_{t=0}^{0}\sum_{s',a'}\gamma^t p(s,a \to s',a';t,\pi)\left(\frac{dr(s',a')}{dx} + \gamma\sum_{s''}\frac{dP(s'';s',a')}{dx}V^\pi_\lambda(s'')\right) \\
&+ \gamma^1 \sum_{\tilde{s},\tilde{a}} p(s,a \to \tilde{s},\tilde{a};1,\pi)\frac{dQ^\pi_\lambda(\tilde{s},\tilde{a})}{dx}.
\end{aligned}$$

We use the definition of $Q^\pi_\lambda(s,a)$ in the first equality. The second follows by the product rule. The third equality follows by the definition of the value function. The last equality comes from rearranging terms.

Induction step $(n \implies n+1)$ Assuming eq. (10) holds for n we show it holds for $n+1$:

$$\begin{aligned}
\frac{dQ^\pi_\lambda(s,a)}{dx} =& \sum_{t=0}^{n}\sum_{s',a'}\gamma^t p(s,a \to s',a';t,\pi)\left(\frac{dr(s',a')}{dx} + \gamma\sum_{s''}\frac{dP(s'';s',a')}{dx}V^\pi_\lambda(s'')\right) \\
&+ \gamma^{n+1}\sum_{\tilde{s},\tilde{a}} p(s,a \to \tilde{s},\tilde{a};n+1,\pi)\frac{dQ^\pi_\lambda(\tilde{s},\tilde{a})}{dx} \\
=& \sum_{t=0}^{n}\sum_{s',a'}\gamma^t p(s,a \to s',a';t,\pi)\left(\frac{dr(s',a')}{dx} + \gamma\sum_{s''}\frac{dP(s'';s',a')}{dx}V^\pi_\lambda(s'')\right) \\
&+ \gamma^{n+1}\sum_{\tilde{s},\tilde{a}} p(s,a \to \tilde{s},\tilde{a};n+1,\pi)\frac{d}{dx}\left(r(\tilde{s},\tilde{a}) + \gamma\sum_{\tilde{s}'}P(\tilde{s}';\tilde{s},\tilde{a})V_\lambda(\tilde{s}')\right) \\
=& \sum_{t=0}^{n}\sum_{s',a'}\gamma^t p(s,a \to s',a';t,\pi)\left(\frac{dr(s',a')}{dx} + \gamma\sum_{s''}\frac{dP(s'';s',a')}{dx}V^\pi_\lambda(s'')\right) \\
&+ \gamma^{n+1}\sum_{\tilde{s},\tilde{a}} p(s,a \to \tilde{s},\tilde{a};n+1,\pi)\left(\frac{d}{dx}r(\tilde{s},\tilde{a}) + \gamma\sum_{\tilde{s}'}\frac{d}{dx}P(\tilde{s}';\tilde{s},\tilde{a})V_\lambda(\tilde{s}')\right. \\
&\left.+ P(\tilde{s}';\tilde{s},\tilde{a})\sum_{\tilde{a}'}\pi(\tilde{a}';\tilde{s}')\frac{d}{dx}Q_\lambda(\tilde{s}',\tilde{a}')\right) \\
=& \sum_{t=0}^{n+1}\sum_{s',a'}\gamma^t p(s,a \to s',a';t,\pi)\left(\frac{dr(s',a')}{dx} + \gamma\sum_{s''}\frac{dP(s'';s',a')}{dx}V^\pi_\lambda(s'')\right) \\
&+ \gamma^{n+2}\sum_{\tilde{s},\tilde{a}} p(s,a \to \tilde{s},\tilde{a};n+2,\pi)\frac{dQ^\pi_\lambda(\tilde{s},\tilde{a})}{dx}.
\end{aligned}$$

The first equality is simply the base case. The second equality follows from the definition of the Q-function. The third equality follows from the multivariate chain rule and the definition of the value function and the last equality is again rearranging terms. □

E.4 Proof of Theorem 3

Proof. By the smoothness of f, we use the following bound from [37][Lemma 1] for $\alpha \leq 1/(2S_f)$:

$$\mathbb{E}\left[\left\|\frac{dF(\hat{x}_T)}{dx}\right\|_\infty^2\right] \leq \underbrace{\frac{2(F(x_1)-\min_x F(x))}{\alpha T}}_{\textbf{(1)}} + \frac{2}{T}\sum_{t=1}^{T}\Bigg(\underbrace{L_f\left\|\mathbb{E}\left[\frac{dF(x_t)}{dx}-\widehat{\frac{dF(x_t)}{dx}}\right]\right\|_\infty}_{\textbf{(2)}} \tag{11}$$

$$+ \underbrace{S_f\alpha\,\mathbb{E}\left[\left\|\frac{dF(x_t)}{dx}-\widehat{\frac{dF(x_t)}{dx}}\right\|_\infty^2\right]}_{\textbf{(3)}}\Bigg).$$

The error term naturally decomposes into an initial error divided by T **(1)**, a bias term **(2)**, and a variance term **(3)**, which decreases with the stepsize α.

For **(1)** we do not need to simplify any further.

To prove our claim, we need to show that **(3)** is $\mathcal{O}(1)$, and that **(2)** is $\mathcal{O}(\delta)$. The latter is the main challenge of this proof.

Let us begin by bounding the bias term **(2)**. The goal is to show that it can be upper bounded by a sum of terms, which are all linear in $\left\|\pi^o_{x,\xi}-\pi^*_{x,\xi}\right\|_\infty$.

$$\begin{aligned}
&\left\|\mathbb{E}\left[\frac{dF(x_t)}{dx}-\widehat{\frac{dF(x_t)}{dx}}\right]\right\|_\infty \\
&= \left\|\mathbb{E}_{x_t}\left[\frac{dF(x_t)}{dx}-\mathbb{E}_{\xi,o}\left[\frac{\partial_1 f(x_t,\pi^o_{x_t,\xi},\xi)}{\partial x}+\mathbb{E}^{\nu}_{a\sim\pi^o_{x_t,\xi}}\left[\frac{1}{\lambda\nu(s)}\frac{\partial_2 f(x_t,\pi^o_{x_t,\xi},\xi))}{\partial\pi(s,a)}\mathbb{E}\left[\widehat{\frac{d}{dx}A^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}}(s,a)\right]\right]\right]\right]\right\|_\infty \\
&\leq \underbrace{\left\|\mathbb{E}_{x_t,o,\xi}\left[\frac{\partial_1 f(x_t,\pi^*_{x_t,\xi},\xi)}{\partial x}-\frac{\partial_1 f(x_t,\pi^o_{x_t,\xi},\xi)}{\partial x}\right]\right\|_\infty}_{\textbf{(A)}} \\
&\quad + \underbrace{\left\|\mathbb{E}^{\xi,\nu}_{x_t,o}\left[\frac{1}{\lambda\nu(s)}\sum_a\left(\pi^*_{x,\xi}(a;s)\frac{\partial_2 f(x,\pi^*_{x,\xi},\xi)}{\partial\pi^*_{x,\xi}(a;s)}\partial_x A^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a)-\pi^o_{x,\xi}(a;s)\frac{\partial_2 f(x,\pi^o_{x_t,\xi},\xi)}{\partial\pi^o_{x_t,\xi}(a;s)}\partial_x A^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s,a)\right)\right]\right\|_\infty}_{\textbf{(B)}},
\end{aligned}$$

where the first equality is by definition and the first inequality follows from the trinalge inequality and we further use the fact that $\widehat{\frac{d}{dx}A^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}}$ is an unbiased estimator of $\frac{d}{dx}A^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}$ as shown in Proposition 3.

(A) is relatively easy to bound. Indeed by the smoothness of f (Assumption 3.1), it immediately follows that

$$\textbf{(A)} \leq \mathbb{E}_{x_t,o,\xi}\left[S_f\left\|\pi^*_{x_t,\xi}-\pi^o_{x_t,\xi}\right\|_\infty\right] \leq S_f\delta.$$

To bound **(B)** we again use the triangle inequality to decompose:

$$\begin{aligned}
\textbf{(B)} &= \left\|\mathbb{E}^{\xi,\nu}_{x_t,o}\left[\frac{1}{\lambda\nu(s)}\sum_a\left(\pi^*_{x,\xi}(a;s)\frac{\partial_2 f(x,\pi^*_{x,\xi},\xi)}{\partial\pi^*_{x,\xi}(a;s)}\partial_x A^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a)-\pi^o_{x,\xi}(a;s)\frac{\partial_2 f(x,\pi^o_{x_t,\xi},\xi)}{\partial\pi^o_{x_t,\xi}(a;s)}\partial_x A^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s,a)\right)\right]\right\|_\infty \\
&\leq \underbrace{\mathbb{E}^{\xi,\nu}_{x_t,o}\left[\frac{1}{\lambda\nu(s)}\sum_a\left\|\pi^*_{x,\xi}(a;s)-\pi^o_{x,\xi}(a;s)\right\|_\infty\left\|\frac{\partial_2 f(x,\pi^*_{x,\xi},\xi)}{\partial\pi^*_{x,\xi}(a;s)}\partial_x A^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty\right]}_{\textbf{(a)}} \\
&\quad + \underbrace{\mathbb{E}^{\xi,\nu}_{x_t,o}\left[\frac{1}{\lambda\nu(s)}\sum_a\left\|\pi^o_{x,\xi}(a;s)\right\|_\infty\left\|\frac{\partial_2 f(x,\pi^*_{x_t,\xi},\xi)}{\partial\pi^*_{x_t,\xi}(a;s)}\partial_x A^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s,a)-\frac{\partial_2 f(x,\pi^o_{x_t,\xi},\xi)}{\partial\pi^o_{x_t,\xi}(a;s)}\partial_x A^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty\right]}_{\textbf{(b)}}.
\end{aligned}$$

(a) is relatively easy to bound. We have

$$\begin{aligned}
\text{(a)} &\leq \mathbb{E}^{\xi,\nu}_{x_t}\left[\frac{1}{\lambda\nu(s)}|\mathcal{A}|\delta\left\|\frac{\partial_2 f(x,\pi^*_{x,\xi},\xi)}{\partial\pi^*_{x,\xi}(a;s)}\partial_x A^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty\right] \\
&\leq \mathbb{E}^{\xi,\nu}_{x_t}\left[\frac{1}{\lambda\nu(s)}|\mathcal{A}|\delta L_f\left\|\partial_x A^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty\right] \\
&\leq \mathbb{E}^{\xi,\nu}_{x_t}\left[\frac{2}{\lambda\nu(s)}|\mathcal{A}|\delta L_f\left\|\partial_x Q^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty\right],
\end{aligned}$$

where we use the assumption on the oracle in the first inequality, that f is Lipschitz continuous in the second inequality, and that $\left\|\partial_x V^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s)\right\|_\infty \leq \left\|\partial_x Q^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty$ in the third inequality. Note that:

$$\left\|V^{\pi}_{\lambda,x,\xi}(s)\right\|_\infty \leq \frac{(\overline{R}+\lambda\log|\mathcal{A}|)}{1-\gamma},$$

since the entropy of any policy is bound by $\log|A|$ (which follows from Jensen's inequality). Using the definition that

$$\partial_x Q^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a) = \mathbb{E}^{\pi^*_{x,\xi}}_{s,a}\left[\sum_{t=0}^{\infty}\gamma^t\frac{dr_{x,\xi}(s_t,a_t)}{dx}+\gamma^{t+1}\frac{d\log P_{x,\xi}(s_{t+1};s_t,a_t)}{dx}V^{\pi}_{\lambda,x,\xi}(s_{t+1})\right],$$

it thus holds that

$$\left\|\partial_x Q^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty \leq \left(\frac{K_2}{1-\gamma}+\frac{K_1(\overline{R}+\lambda\log|\mathcal{A}|)}{(1-\gamma)^2}\right). \tag{12}$$

Letting $m := \min_s \nu(s)$, we thus have

$$\text{(a)} \leq \frac{2}{\lambda m}|\mathcal{A}|\delta L_f\left(\frac{K_2}{1-\gamma}+\frac{K_1(\overline{R}+\lambda\log|\mathcal{A}|)}{(1-\gamma)^2}\right).$$

For (b) we further simplify using the triangle inequallity:

$$\begin{aligned}
\text{(b)} &\leq \frac{1}{\lambda m}\mathbb{E}^{\xi}_{x_t,o}\left[\left\|\frac{\partial_2 f(x,\pi^*_{x_t,\xi},\xi)}{\partial\pi^*_{x_t,\xi}(a;s)}\partial_x A^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s,a)-\frac{\partial_2 f(x,\pi^o_{x_t,\xi},\xi)}{\partial\pi^o_{x_t,\xi}(a;s)}\partial_x A^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty\right] \\
&\leq \underbrace{\frac{1}{\lambda m}\mathbb{E}^{\xi}_{x_t,o}\left[\left\|\frac{\partial_2 f(x,\pi^*_{x_t,\xi},\xi)}{\partial\pi^*_{x_t,\xi}(a;s)}-\frac{\partial_2 f(x,\pi^o_{x_t,\xi},\xi)}{\partial\pi^o_{x_t,\xi}(a;s)}\right\|_\infty\left\|\partial_x A^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty\right]}_{\text{(i)}} \\
&\quad + \underbrace{\frac{1}{\lambda m}\mathbb{E}^{\xi}_{x_t,o}\left[\left\|\frac{\partial_2 f(x,\pi^o_{x_t,\xi},\xi)}{\partial\pi^o_{x_t,\xi}(a;s)}\right\|_\infty\left\|\partial_x A^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s,a)-\partial_x A^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty\right]}_{\text{(ii)}}.
\end{aligned}$$

Similar to (a), we can bound (i) using the smoothness of f (Assumption 3.1):

$$\text{(i)} \leq 2\frac{S_f}{\lambda m}\delta\left(\frac{K_2}{1-\gamma}+\frac{K_1(\overline{R}+\lambda\log|\mathcal{A}|)}{(1-\gamma)^2}\right).$$

Bounding (ii) and in particular $\left\|\partial_x A^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s,a)-\partial_x A^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty$ is the tricky part of this proof. We first need to show two intermediate results. First, we bound the difference in entropy between two policies and the difference in the regularized value functions of two policies. Once we have

that, we can tackle $\left\|\partial_x A^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s,a) - \partial_x A^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty$. For the entropy, we denote by $l_1 := \min_{s,a,x,\xi} \pi^*_{x,\xi}(a;s)$ the minimum probability of playing an action in any state under the optimal policy. Recall that $\forall x, \xi, s, a : |r_{x,\xi}(s,a)| < \overline{R}$ and that $0 \leq H(\pi; s) \leq \log|\mathcal{A}|$ (by Jensen's inequality). Thus we have

$$\frac{-\overline{R}}{1-\gamma} \leq Q^*_{\lambda,x,\xi}(s,a) \leq \frac{\overline{R} + \lambda \log|\mathcal{A}|}{1-\gamma}.$$

Because $\pi^*_{x,\xi}(s;a) \propto \exp(Q^*_{\lambda,x,\xi}(s,a)/\lambda)$, it follows that

$$l_1 \geq \frac{\exp(\frac{-\overline{R}}{\lambda(1-\gamma)})}{|\mathcal{A}|\exp(\frac{\overline{R}+\lambda\log|\mathcal{A}|}{\lambda(1-\gamma)})} > 0.$$

We assume now that δ is sufficiently small, i.e. $\delta \leq l_1/2$, such that $l_1/2 \leq \min_{s,a,x,\xi} \pi^o_{x,\xi}(a;s)$. We need to have such a lower bound on the policies, touse the fact that the $\log$ function is Lipschitz continuous with parameter $\frac{1}{a}$ on an interval $[a,\infty)$ for any $a > 0$. Hence we have

$$\begin{aligned}
\left\|H(\pi^*_{x,\xi}|s) - H(\pi^o_{x,\xi}|s)\right\|_\infty &= \left\|\sum_a \pi^*_{x,\xi}(a;s)\log\pi^*_{x,\xi}(a;s) - \sum_a \pi^o_{x,\xi}(a;s)\log\pi^o_{x,\xi}(a;s)\right\|_\infty \\
&\leq \left(\sum_a \left\|\pi^*_{x,\xi} - \pi^o_{x,\xi}\right\|_\infty \left\|\log\pi^*_{x,\xi}\right\|_\infty\right) + \left\|\log\pi^*_{x,\xi} - \log\pi^o_{x,\xi}\right\|_\infty \\
&\leq |\mathcal{A}||\log l_1|\delta + \frac{2}{l_1}\delta.
\end{aligned}$$

We use this to bound the difference in the value functions of $\pi^*_{x,\xi}$ and $\pi^o_{x,\xi}$.

$$\begin{aligned}
&\left\|V^{\pi^*_{x,\xi}}_\lambda(s) - V^{\pi^o_{x,\xi}}_\lambda(s)\right\|_\infty \\
\leq &\left\|\sum_a \left(\pi^*_{x,\xi}(a;s)Q^{\pi^*_{x,\xi}}_\lambda(s,a) - \pi^o_{x,\xi}(a;s)Q^{\pi^o_{x,\xi}}_\lambda(s,a)\right)\right\|_\infty + \lambda\left\|H(\pi^*_{x,\xi}|s) - H(\pi^o_{x,\xi}|s)\right\|_\infty \\
\leq &\left\|\sum_a \left(\pi^*_{x,\xi}(a;s)Q^{\pi^*_{x,\xi}}_\lambda(s,a) - \pi^o_{x,\xi}(a;s)Q^{\pi^o_{x,\xi}}_\lambda(s,a)\right)\right\|_\infty + \lambda\delta\left(|\mathcal{A}||\log l_1| + \frac{2}{l_1}\right) \\
\leq &\left\|\sum_a \pi^*_{x,\xi}(a;s)\right\|_\infty \left\|Q^{\pi^*_{x,\xi}}_\lambda(s,a) - Q^{\pi^o_{x,\xi}}_\lambda(s,a)\right\|_\infty \\
&+ \sum_a \left\|\pi^*_{x,\xi}(a;s) - \pi^o_{x,\xi}(a;s)\right\|_\infty \left\|Q^{\pi^o_{x,\xi}}_\lambda(s,a)\right\|_\infty + \lambda\delta\left(|\mathcal{A}||\log l_1| + \frac{2}{l_1}\right) \\
\leq &\lambda\delta\left(|\mathcal{A}||\log l_1| + \frac{2}{l_1}\right) + \delta|\mathcal{A}|\frac{\overline{R}}{1-\gamma} + \gamma\left\|V^{\pi^*_{x,\xi}}_\lambda(s) - V^{\pi^o_{x,\xi}}_\lambda(s)\right\|_\infty \\
\leq &\frac{\lambda\delta\left(|\mathcal{A}||\log l_1| + \frac{2}{l_1}\right)}{1-\gamma} + \frac{\delta|\mathcal{A}|\overline{R}}{(1-\gamma)^2}.
\end{aligned} \tag{13}$$

The first inequality follows from the definition of the regularized value function and the triangle inequality. The second inequality follows from our derived bound on the difference of entropies. The third inequality is agian using the triangle inequality and the fourth inequality is bounding the Q-function using that the rewards are upper bounded by $\overline{R}$ and that $Q^\pi_{\lambda,x,\xi}(s,a) = r_{x,\xi}(s,a) + \gamma\mathbb{E}_{s'\sim P_{x,\xi}(\cdot;s,a)}\left[V^\pi_{\lambda,x,\xi}(s')\right]$. The last inequality then follows by iteratively plugging in the inequality for the term $\left\|V^{\pi^*_{x,\xi}}_\lambda(s) - V^{\pi^o_{x,\xi}}_\lambda(s)\right\|_\infty$, which gives a geometric sum. We employ a similar technique to bound **(ii)** using the above results:

$$
\begin{aligned}
&\frac{1}{\lambda m}\mathbb{E}^{\xi}_{x_t,o}\left[\left\|\frac{\partial_2 f(x,\pi^o_{x_t,\xi},\xi)}{\partial \pi^o_{x_t,\xi}(a;s)}\right\|_\infty \left\|\partial_x A^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s,a)-\partial_x A^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty\right]\\
\leq&\frac{L_f}{\lambda m}\mathbb{E}^{\xi}_{x_t,o}\left[\left\|\partial_x A^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s,a)-\partial_x A^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty\right]\\
\leq&\frac{L_f}{\lambda m}2\mathbb{E}^{\xi}_{x_t,o}\left[\left\|\partial_x Q^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s,a)-\partial_x Q^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty+|\mathcal{A}|\left(\frac{K_2}{1-\gamma}+\frac{K_1(\overline{R}+\lambda\log|\mathcal{A}|)}{(1-\gamma)^2}\right)\left\|\pi^o_{x,\xi}-\pi^*_{x,\xi}\right\|_\infty\right].
\end{aligned}
$$

Here the first inequality follows from the Lipschitz continuity of f. For second inequality we use the definition of the advantage function, the triangle inequality and the following inequality:

$$
\begin{aligned}
&\left\|\partial_x V^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s)-\partial_x V^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s)\right\|_\infty\\
=&\left\|\sum_a \pi^o_{x_t,\xi}(a;s)\partial_x Q^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s,a)-\sum_a \pi^*_{x_t,\xi}(a;s)\partial_x Q^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty\\
=&\left\|\sum_a\left(\pi^o_{x_t,\xi}(a;s)-\pi^*_{x_t,\xi}(a;s)\right)\partial_x Q^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s,a)-\sum_a \pi^*_{x_t,\xi}(a;s)\left(\partial_x Q^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s,a)-\partial_x Q^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s,a)\right)\right\|_\infty\\
\leq&|\mathcal{A}|\frac{K_1(\overline{R}+\lambda\log|\mathcal{A}|)}{(1-\gamma)^2}\left\|\pi^o_{x_t,\xi}-\pi^*_{x_t,\xi}\right\|_\infty+\left\|\partial_x Q^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s,a)-\partial_x Q^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty,
\end{aligned}
\tag{14}
$$

where the last inequality follows from the trianlge inequality and Equation (12).

We bound the difference in Q-function derivatives as follows:

$$
\begin{aligned}
&\left\|\partial_x Q^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s,a)-\partial_x Q^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s,a)\right\|_\infty\\
=&\Bigg\|\sum_{t=0}^{\infty}\sum_{s',a'}\gamma^t p_{x,\xi}(s,a\to s',a';t,\pi^*_{x_t,\xi})\left(\frac{dr_{x,\xi}(s',a')}{dx}+\gamma\sum_{s''}\frac{dP_{x,\xi}(s'';s',a')}{dx}V^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s'')\right)\\
&-\sum_{t=0}^{\infty}\sum_{s',a'}\gamma^t p_{x,\xi}(s,a\to s',a';t,\pi^o_{x_t,\xi})\left(\frac{dr_{x,\xi}(s',a')}{dx}+\gamma\sum_{s''}\frac{dP_{x,\xi}(s'';s',a')}{dx}V^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s'')\right)\Bigg\|_\infty\\
=&\Bigg\|\frac{dr_{x,\xi}(s,a)}{dx}+\gamma\sum_{s'}\frac{dP_{x,\xi}(s';s,a)}{dx}V^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s')\\
&+\sum_{t=1}^{\infty}\sum_{s',a'}\gamma^t p_{x,\xi}(s,a\to s',a';t,\pi^*_{x_t,\xi})\left(\frac{dr_{x,\xi}(s',a')}{dx}+\gamma\sum_{s''}\frac{dP_{x,\xi}(s'';s',a')}{dx}V^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s'')\right)\\
&-\frac{dr_{x,\xi}(s,a)}{dx}-\gamma\sum_{s'}\frac{dP_{x,\xi}(s';s,a)}{dx}V^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s')\\
&-\sum_{t=1}^{\infty}\sum_{s',a'}\gamma^t p_{x,\xi}(s,a\to s',a';t,\pi^o_{x_t,\xi})\left(\frac{dr_{x,\xi}(s',a')}{dx}+\gamma\sum_{s''}\frac{dP_{x,\xi}(s'';s',a')}{dx}V^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s'')\right)\Bigg\|_\infty\\
\leq&\gamma\sum_{s'}\left\|\frac{dP_{x,\xi}(s';s,a)}{dx}\right\|_\infty\left\|V^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s')-V^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s')\right\|_\infty\\
&+\gamma\Bigg\|\sum_{s',a'}P(s';s,a)\pi^*_{x_t,\xi}(a',s')\sum_{t=0}^{\infty}\sum_{s',a'}\gamma^t p_{x,\xi}(s',a'\to s'',a'';t,\pi^*_{x_t,\xi})\ldots\\
&\left(\frac{dr_{x,\xi}(s'',a'')}{dx}+\gamma\sum_{s'''}\frac{dP_{x,\xi}(s''';s'',a'')}{dx}V^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s'')\right)\\
&-\sum_{s',a'}P(s';s,a)\pi^o_{x_t,\xi}(a',s')\sum_{t=0}^{\infty}\sum_{s',a'}\gamma^t p_{x,\xi}(s',a'\to s'',a'';t,\pi^o_{x_t,\xi})\ldots
\end{aligned}
$$

$$
\begin{aligned}
&\left(\frac{dr_{x,\xi}(s'',a'')}{dx} + \gamma \sum_{s'''} \frac{dP_{x,\xi}(s'''; s'', a'')}{dx} V^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s'')\right)\Bigg|\Bigg|_\infty \\
\leq &\gamma \sum_{s'} \left|\left|\frac{dP_{x,\xi}(s'; s, a)}{dx}\right|\right|_\infty \left|\left|V^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s') - V^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s')\right|\right|_\infty \\
&+ \gamma \Bigg|\Bigg| \sum_{s',a'} P(s'; s, a)\pi^*_{x_t,\xi}(a', s')\partial_x Q^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s, a) \\
&- \sum_{s',a'} P(s'; s, a)\pi^o_{x_t,\xi}(a', s')\partial_x Q^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s, a)\Bigg|\Bigg|_\infty \\
\leq &\gamma \sum_{s'} \left|\left|\frac{dP_{x,\xi}(s'; s, a)}{dx}\right|\right|_\infty \left|\left|V^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s') - V^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s')\right|\right|_\infty \\
&+ \gamma \left|\left|\partial_x Q^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s', a')\right|\right|_\infty \left|\left|\sum_{s',a'} P(s'; s, a)\right|\right|_\infty \left|\left|\pi^*_{x_t,\xi}(a', s') - \pi^o_{x_t,\xi}(a', s')\right|\right|_\infty \\
&+ \gamma \sum_{s',a'} P(s'; s, a)\pi^o_{x_t,\xi}(a', s') \left|\left|\partial_x Q^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s', a') - \partial_x Q^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s', a')\right|\right|_\infty,
\end{aligned}
$$

where the dots indicate multiplication over the linebreak. The first equality follows from plugging in the result from Theorem 2. The second equality follows by taking out all terms with $t = 0$. The first inequality uses the triangle inequality. The second inequality plugs back in the definition from Theorem 2. The last inequality follows from the triangle inequality again.

Taking the expectation, we thus get:

$$
\begin{aligned}
&\mathbb{E}^{\xi}_{x_t,o}\left[\left|\left|\partial_x Q^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s, a) - \partial_x Q^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s, a)\right|\right|_\infty\right] \\
\leq &\gamma \left(|\mathcal{S}|K_1 \left(\frac{\lambda\delta\left(|\mathcal{A}||\log l_1| + \frac{2}{l_1}\right)}{1-\gamma} + \frac{\delta|\mathcal{A}|\overline{R}}{(1-\gamma)^2}\right) + \delta\left(\frac{K_2}{1-\gamma} + \frac{K_1(\overline{R} + \lambda\log|\mathcal{A}|)}{(1-\gamma)^2}\right)\right) \\
&+ \gamma\mathbb{E}^{\xi}_{x_t,o}\left[\left|\left|\partial_x Q^{\pi^*_{x_t,\xi}}_{\lambda,x,\xi}(s', a') - \partial_x Q^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}(s', a')\right|\right|_\infty\right] \\
\leq &\frac{\gamma}{1-\gamma}\left(|\mathcal{S}|K_1 \left(\frac{\lambda\delta\left(|\mathcal{A}||\log l_1| + \frac{2}{l_1}\right)}{1-\gamma} + \frac{\delta|\mathcal{A}|\overline{R}}{(1-\gamma)^2}\right) + \delta\left(\frac{K_2}{1-\gamma} + \frac{K_1(\overline{R} + \lambda\log|\mathcal{A}|)}{(1-\gamma)^2}\right)\right) \\
= &\frac{\delta\gamma}{1-\gamma}\left(|\mathcal{S}|K_1 \left(\frac{\lambda\left(|\mathcal{A}||\log l_1| + \frac{2}{l_1}\right)}{1-\gamma} + \frac{|\mathcal{A}|\overline{R}}{(1-\gamma)^2}\right) + \left(\frac{K_2}{1-\gamma} + \frac{K_1(\overline{R} + \lambda\log|\mathcal{A}|)}{(1-\gamma)^2}\right)\right),
\end{aligned}
$$

where we use the intermediate result from before to bound the difference between the value functions, the upper bound on the Q-function derivative shown in Equation (12) and the assumption on the oracle to get the first inequality. The second inequality follows from the the resulting geometric sum and the last equality is just rearranging terms to show the linearity in δ.

Using this result, we can now bound **(ii)**:

$$
\begin{aligned}
\textbf{(ii)} \leq &\frac{2L_f\delta\gamma}{\lambda m(1-\gamma)}\left(|\mathcal{S}|K_1 \left(\frac{\lambda\left(|\mathcal{A}||\log l_1| + \frac{2}{l_1}\right)}{1-\gamma} + \frac{|\mathcal{A}|\overline{R}}{(1-\gamma)^2}\right) + \left(\frac{K_2}{1-\gamma} + \frac{K_1(\overline{R} + \lambda\log|\mathcal{A}|)}{(1-\gamma)^2}\right)\right. \\
&\left. + |\mathcal{A}|\left(\frac{K_2}{\gamma} + \frac{K_1(\overline{R} + \lambda\log|\mathcal{A}|)}{(1-\gamma)\gamma}\right)\right).
\end{aligned}
$$

With that we are done decomposing **(2)**. Combining everything, we have the following bound:

$$
\textbf{(2)} = \left|\left|\mathbb{E}\left[\frac{dF(x_t)}{dx} - \widehat{\frac{dF(x_t)}{dx}}\right]\right|\right|_\infty
$$

$$
\begin{aligned}
&\leq \textbf{(A)} + \textbf{(B)} \\
&\leq \textbf{(A)} + \textbf{(a)} + \textbf{(b)} \\
&\leq \textbf{(A)} + \textbf{(a)} + \textbf{(i)} + \textbf{(ii)} \\
&\leq S_f \delta + \frac{2}{\lambda m} |\mathcal{A}| \delta L_f \left(\frac{K_2}{1-\gamma} + \frac{K_1(\overline{R} + \lambda \log |\mathcal{A}|)}{(1-\gamma)^2} \right) + \frac{2S_f}{\lambda m} \delta \left(\frac{K_2}{1-\gamma} + \frac{K_1(\overline{R} + \lambda \log |\mathcal{A}|)}{(1-\gamma)^2} \right) \\
&\quad + \frac{2L_f \delta \gamma}{\lambda m (1-\gamma)} \left(|\mathcal{S}| K_1 \left(\frac{\lambda \left(|\mathcal{A}| |\log l_1| + \frac{2}{l_1} \right)}{1-\gamma} + \frac{|\mathcal{A}| \overline{R}}{(1-\gamma)^2} \right) + \left(\frac{K_2}{1-\gamma} + \frac{K_1(\overline{R} + \lambda \log |\mathcal{A}|)}{(1-\gamma)^2} \right) \right. \\
&\quad \left. + |\mathcal{A}| \left(\frac{K_2}{\gamma} + \frac{K_1(\overline{R} + \lambda \log |\mathcal{A}|)}{(1-\gamma)\gamma} \right) \right) \\
&= \mathcal{O}(\delta).
\end{aligned}
$$

With that we have tackled terms **(1)** and **(2)** in Equation (11). It remains to bound the variance, i.e. term **(3)**. If we can show that **(3)** $= \mathcal{O}(1)$ then the last term in Equation (11) is $\mathcal{O}(\alpha)$ as claimed and we are done. Indeed, bounding **(3)** is relatively easy, as all important terms are bounded by Assumption 3.1. We have:

$$
\begin{aligned}
\mathbb{E}\left[\left\| \frac{dF(x_t)}{dx} - \widehat{\frac{dF(x_t)}{dx}} \right\|_\infty^2 \right] &\leq 2 \left\| \frac{dF(x_t)}{dx} - \mathbb{E}\left[\widehat{\frac{dF(x_t)}{dx}} \right] \right\|_\infty^2 + 2\mathbb{E}\left[\left\| \widehat{\frac{dF(x_t)}{dx}} - \mathbb{E}\left[\widehat{\frac{dF(x_t)}{dx}} \right] \right\|_\infty^2 \right] \\
&\leq 2\mathcal{O}(\delta^2) + 2\mathbb{E}\left[\left\| \widehat{\frac{dF(x_t)}{dx}} - \mathbb{E}\left[\widehat{\frac{dF(x_t)}{dx}} \right] \right\|_\infty^2 \right] \\
&\leq \mathcal{O}(\delta^2) + 2\left(\mathbb{E}\left[\left\| \widehat{\frac{dF(x_t)}{dx}} \right\|_\infty^2 \right] - \left\| \mathbb{E}\left[\widehat{\frac{dF(x_t)}{dx}} \right] \right\|_\infty^2 \right) \\
&\leq \mathcal{O}(\delta^2) + 2\mathbb{E}\left[\left\| \widehat{\frac{dF(x_t)}{dx}} \right\|_\infty^2 \right],
\end{aligned}
$$

where the third inequality follows directly, the second inequality uses the definition of the variance and the first inequality uses that

$$
\|a+b\|^2 = \|a\|^2 + 2a^\top b + \|b\|^2 \leq \|a\|^2 + 2\|a\|\|b\| + \|b\|^2 \leq 2\|a\| + 2\|b\|^2 .
$$

By the above, it suffices to bound the second moment:

$$
\begin{aligned}
\mathbb{E}\left[\left\| \widehat{\frac{dF(x_t)}{dx}} \right\|_\infty^2 \right] &\leq \mathbb{E}\left[\left\| \frac{\partial_1 f(x_t, \pi^o_{x_t,\xi}, \xi)}{\partial x} + \frac{1}{\lambda \nu(s)} \frac{\partial_2 f(x_t, \pi^o_{x_t,\xi}, \xi))}{\partial \pi(s,a)} \widehat{\partial_x A^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}}(s,a) \right\|_\infty^2 \right] \\
&\leq \mathbb{E}\left[\left\| L_f + \frac{1}{\lambda m} L_f \widehat{\partial_x A^{\pi^o_{x_t,\xi}}_{\lambda,x,\xi}}(s,a) \right\|_\infty^2 \right].
\end{aligned}
$$

To proceed, we upper bound $\mathbb{E}\left[\left\|\widehat{\partial_x A^{\pi^o_{x_t},\xi}_{\lambda,x,\xi}}(s,a)\right\|^2_\infty\right]$ by

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\widehat{\partial_x A^{\pi^o_{x_t},\xi}_{\lambda,x,\xi}}(s,a)\right\|^2_\infty\right] \\
\leq& 4\mathbb{E}\Bigg[\Bigg\|\Bigg(\sum_{t=0}^{T_Q}\frac{d}{dx}r(s_t^{\tau_Q},a_t^{\tau_Q}) \\
&+\frac{\gamma}{1-\gamma}\frac{d}{dx}\log P(s^{\tau_Q}_{T_Q+1};s^{\tau_Q}_{T_Q},a^{\tau_Q}_{T_Q})\sum_{t=T_Q+1}^{T_Q+T'_Q+1}\gamma^{(t-T_Q-1)/2}\left(r(s_t^{\tau_Q},a_t^{\tau_Q})+\lambda H(\pi(\cdot;s_t))\right)\Bigg)\Bigg\|^2_\infty\Bigg] \\
\leq& 4\mathbb{E}\left[\left\|\left(T_Q K_2+\frac{\gamma}{1-\gamma}K_1\frac{\overline{R}+\lambda\log|\mathcal{A}|}{1-\gamma^{0.5}}\right)\right\|^2_\infty\right],
\end{aligned}
\tag{15}
$$

where T_Q is the random variable defined in Algorithm 2. The first inequality uses the definition of the advantage estimate (cf. Algorithm 2), the i.i.d. property of T_Q and T_V, as well as of T'_Q and T'_V, and Equation (16). The second inequality follows from Assumption 3.1.

Plugging in the above, we thus get:

$$
\begin{aligned}
\mathbb{E}\left[\left\|\widehat{\frac{dF(x_t)}{dx}}\right\|^2_\infty\right] &= \mathbb{E}\left[\left\|L_f+\frac{1}{\lambda m}L_f\widehat{\partial_x A^{\pi^o_{x_t},\xi}_{\lambda,x,\xi}}(s,a)\right\|^2_\infty\right] \\
&\leq \mathbb{E}\left[2\left\|L_f\right\|^2_\infty+2\left(\frac{L_f}{\lambda m}\right)^2\left\|\widehat{\partial_x A^{\pi^o_{x_t},\xi}_{\lambda,x,\xi}}(s,a)\right\|^2_\infty\right] \\
&\leq 2\mathbb{E}\left[\left\|L_f\right\|^2_\infty\right]+8\left(\frac{L_f}{\lambda m}\right)^2\mathbb{E}\left[\left\|\left(T_Q K_2+\frac{\gamma}{1-\gamma}K_1\frac{\overline{R}+\lambda\log|\mathcal{A}|}{1-\gamma^{0.5}}\right)\right\|^2_\infty\right] \\
&\leq 2\mathbb{E}\left[\left\|L_f\right\|^2_\infty\right]+16\left(\frac{L_f}{\lambda m}\right)^2\mathbb{E}\left[\left\|T_Q K_2\right\|^2_\infty\right]+16\mathbb{E}\left[\left\|\frac{\gamma}{1-\gamma}K_1\frac{\overline{R}+\lambda\log|\mathcal{A}|}{1-\gamma^{0.5}}\right\|^2_\infty\right] \\
&= \mathcal{O}(1),
\end{aligned}
$$

where we repeatedly apply Equation (16) and the fact that the second moment of a geometric random variable is finite.

Now we can plug all our bounds back into Equation (11) to get the result of Theorem 3.

$$
\begin{aligned}
\mathbb{E}\left[\left\|\frac{dF(\hat{x}_T)}{dx}\right\|^2_\infty\right] &\leq \textbf{(1)}+\textbf{(2)}+\textbf{(3)} \\
&\leq \mathcal{O}(\frac{1}{\alpha T})+\mathcal{O}(\delta)+2S_f\alpha\left(\mathcal{O}(\delta^2+1)\right) \\
&\leq \mathcal{O}(\frac{1}{\alpha T})+\mathcal{O}(\delta)+\mathcal{O}(\alpha).
\end{aligned}
$$

□

E.5 Proof of Theorem 4

Vanilla soft Q-learning

We give a brief overview of how HPGD is combined with vanilla soft Q-learning (Algorithm 4) to get a bias of $\mathcal{O}(2^{-K/2})$ in Algorithm 7. In the algorithm we refer to $t_K = \mathcal{O}(K2^K)$ as the number of iterations soft Q-learning needs to achieve $\lambda\mathbb{E}\|\pi^{t_K}-\pi^*\|^2_\infty \leq \mathbb{E}\|Q^{t_K}-Q^*_\lambda\|^2_\infty \leq 2^{-K}$ (cf. Proposition 6). Note we slightly abuse notation, when we pass a policy instead of an oracle

to Algorithm 2. However, the policy can be equivalently used to sample trajectories. Note that the idea of HPGD with soft Q-learning is a double-loop methods that aims to find an δ-oracle. Similar a δ-oracle idea also appears in conditional stochastic optimization [38, 35], in bilevel optimization [28].

Algorithm 7 HPGD with vanilla soft Q-learning

Input: Iterations T, Precision param. K, Learning rate α, Regularization λ, Initial point x_0, behavioural policy π_B, Q-learning rates $\{\alpha_t\}_{t\geq 0}$
for $t = 0$ to $T-1$ **do**
  $\xi \sim \mathbb{P}_\xi$
  $\pi^{t_K} \leftarrow \texttt{SoftQlearning}_{x_t,\xi}(t_K, \pi_B, \{\alpha_t\}_{t\geq 0})$ (Algorithm 4)
  $s \sim \nu$ and $a \sim \pi^{t_K}(\cdot; s)$
  $\widehat{\partial_x A^{\pi^{t_K}}}(s,a) \leftarrow \texttt{GradientEstimator}(\xi, x_t, s, a, \pi^{t_K})$ (Algorithm 2)
  $\widehat{\frac{dF}{dx}} \leftarrow \frac{\partial_1 f(x_t, \pi^{t_K}, \xi)}{\partial x} + \frac{1}{\lambda\nu(s)} \frac{\partial_2 f(x_t, \pi^{t_K}, \xi)}{\partial \pi(s,a)} \widehat{\partial_x A^{\pi^{t_K}}}(s,a)$
  $x_{t+1} \leftarrow x_t - \alpha \widehat{\frac{dF}{dx}}$
end for
Output: $\hat{x}_T \sim \text{Uniform}(\{x_0, \ldots, x_{T-1}\})$

Randomly-Truncated soft Q-learning (RT-Q)

Let us now turn to RT-Q, for which we provide the pseudocode in Algorithm 8. As above, we denote by $t_k = \mathcal{O}(k2^k)$ the number of iterations soft Q-learning needs to achieve $\lambda \mathbb{E}\|\pi^{t_k} - \pi^*\|_\infty^2 \leq \mathbb{E}\|Q^{t_k} - Q_\lambda^*\|_\infty^2 \leq 2^{-k}$. We slightly abuse notation in the Pseudocode, such that we do not just return the last soft Q-learning iteration but also the second last and the first. Moreover we denote by $p_k = \frac{2^{-k}}{1-2^{-K}}$ and we generally use $\widehat{x}$ to denote estimates from a single sample and $\widetilde{x}$ to denote averaged estimates from multiple samples.

Algorithm 8 HPGD with RT-Q

Input: Iterations T, Precision param. K, Learning rate α, Regularization λ, Initial point x_0, behavioural policy π_B, Q-learning rates $\{\alpha_t\}_{t\geq 0}$
for $t = 0$ to $T-1$ **do**
  $\xi \sim \mathbb{P}_\xi$
  $k \sim p_k$
  $\pi^{t_{k+1}}, \pi^{t_k}, \pi^{t_1} \leftarrow \texttt{SoftQlearning}_{x_t,\xi}(t_{k+1}, \pi_B, \{\alpha_t\}_{t\geq 0})$ (Algorithm 4)
  $s \sim \nu$, $a \sim \pi^{t_{k+1}}(\cdot; s)$ and $a' \sim \pi^{t_1}(\cdot; s)$
  $\widetilde{\partial_x A^{\pi^{t_k}}}(s,a) \leftarrow \frac{1}{2^k} \sum_{l=1}^{2^k} \texttt{GradientEstimator}(\xi, x_t, s, a, \pi^{t_k})$(Algorithm 2)
  $\widetilde{\partial_x A^{\pi^{t_{k+1}}}}(s,a) \leftarrow \frac{1}{2^k} \sum_{l=1}^{2^k} \texttt{GradientEstimator}(\xi, x_t, s, a, \pi^{t_{k+1}})$
  $\widehat{\partial_x A^{\pi^{t_1}}}(s,a') \leftarrow \texttt{GradientEstimator}(\xi, x_t, s, a', \pi^{t_1})$
  $\widetilde{\frac{dF_{t_{k+1}}}{dx}} \leftarrow \frac{\partial_1 f(x, \pi^{t_{k+1}}, \xi)}{\partial x} + \frac{1}{\lambda\nu(s)} \frac{\partial_2 f(x, \pi^{t_{k+1}}, \xi))}{\partial \pi(s,a)} \widetilde{\partial_x A^{\pi^{t_{k+1}}}}(s,a)$
  $\widetilde{\frac{dF_{t_k}^{t_{k+1}}}{dx}} = \frac{\partial_1 f(x, \pi^{t_k}, \xi)}{\partial x} + \frac{\pi^{t_k}(a;s)}{\pi^{t_{k+1}}(a;s)} \frac{1}{\lambda\nu(s)} \frac{\partial_2 f(x, \pi^{t_k}, \xi))}{\partial \pi(s,a)} \widetilde{\partial_x A^{\pi^{t_k}}}(s,a)$
  $\widehat{\frac{dF_{t_1}}{dx}} = \frac{\partial_1 f(x, \pi^{t_1}, \xi)}{\partial x} + \frac{1}{\lambda\nu(s)} \frac{\partial_2 f(x, \pi^{t_1}, \xi))}{\partial \pi(s,a')} \widehat{\partial_x A^{\pi^{t_1}}}(s,a')$
  $\frac{dF_{t_K}^{RT}}{dx} = \widehat{\frac{dF_{t_1}}{dx}} + \frac{\widetilde{\frac{dF_{t_{k+1}}}{dx}} - \widetilde{\frac{dF_{t_k}^{t_{k+1}}}{dx}}}{p_k}$
  $x_{t+1} \leftarrow x_t - \alpha \frac{dF_{t_K}^{RT}}{dx}$
end for
Output: $\hat{x}_T \sim \text{Uniform}(\{x_0, \ldots, x_{T-1}\})$

Proof. First let us specify that "bias" refers to the bias of the hypergradient estimator, i.e.

$$\left\| \mathbb{E}\left[\frac{dF(x_t)}{dx} - \widehat{\frac{dF(x_t)}{dx}} \right] \right\|_\infty .$$

Equivalently for "variance" we mean the variance of the estimator i.e.

$$\mathbb{E}\left[\left\| \widehat{\frac{dF(x_t)}{dx}} - \mathbb{E}\left[\widehat{\frac{dF(x_t)}{dx}} \right] \right\|_\infty^2 \right],$$

Moreover, the "iteration complexity" is the number of soft Q-learning iterations needed to solve the lower level.

For Vanilla soft Q-learning, we can just combine previous results to compute the iteration complexity and variance to achieve a bias of $2^{-K/2}$. For RT-Q, we formalize the intuition of Section 4 to show we can achieve the same bias with only $\mathcal{O}(K^2)$ iterations because we rarely perform many soft Q-learning iterations but assign a relatively higher magnitude to these few accurate hypergradient estimates. This necessarily increases variance and most of the proof will be spent on how to bound it. Here we "divide and conquer" the variance until we have easy terms that depend linearly on

$$\mathbb{E}\left[\left\| \pi_{x,\xi}^{t_k} - \pi_{x,\xi}^* \right\|_\infty^2 \right] = \mathcal{O}(2^{-k})$$

or related terms we can easily bound. To decompose the variance, our main tool will be the following identity

$$\|a+b\|^2 = \|a\|^2 + 2\|a\|\|b\| + \|b\|^2 \leq 2\|a\| + 2\|b\|^2, \tag{16}$$

which we have already derived and used in the proof of Theorem 3 and also follows from the Parallelogram Law.

Vanilla soft Q-learning

We start with the analysis of using HPGD with vanilla soft Q-learning to estimate $\frac{dF(x)}{dx}$. In Theorem 3, we showed that

$$\left\| \mathbb{E}\left[\frac{dF(x_t)}{dx} - \widehat{\frac{dF(x_t)}{dx}} \right] \right\|_\infty = \mathcal{O}(\delta),$$

where

$$\mathbb{E}_o\left[\left\| \pi_{x,\xi}^* - \pi_{x,\xi}^o \right\|_\infty^2 \right] \leq \delta^2.$$

In Proposition 6, we show that after $t_K = \mathcal{O}(K2^K)$ soft Q-learning iterations, it holds that

$$\mathbb{E}\left[\left\| \pi_{x,\xi}^{t_K} - \pi_{x,\xi}^* \right\|_\infty^2 \right] = \mathcal{O}(2^{-K}),$$

where $\pi_{x,\xi}^{t_K}$ denotes the t_K-th iterate of the soft Q-learning algorithm. The results for complexity and bias follow directly. It remains to bound the variance. However, we have already shown in Theorem 3 that

$$\mathbb{E}\left[\left\| \widehat{\frac{dF_{t_K}}{dx}} - \mathbb{E}\left[\widehat{\frac{dF_{t_K}}{dx}} \right] \right\|_\infty^2 \right] \leq \mathbb{E}\left[\left\| \widehat{\frac{dF(x_t)}{dx}} \right\|_\infty^2 \right] = \mathcal{O}(1).$$

Randomly-Truncated soft Q-learning (RT-Q)

We first show that $\frac{dF_{t_K}^{RT}}{dx}$ has the same bias as its vanilla counterpart $\widehat{\frac{dF_{t_K}}{dx}}$. For this, observe that the following estimators have the same mean:

$$\forall k : \mathbb{E}\left[\widetilde{\frac{dF_{t_k}^{t_{k+1}}}{dx}} \right] = \mathbb{E}\left[\widetilde{\frac{dF_{t_k}}{dx}} \right] = \mathbb{E}\left[\widehat{\frac{dF_{t_k}}{dx}} \right].$$

The first equality holds because of importance sampling and the second inequality holds by the linearity of expectation. Plugging in these identities, we get

$$
\begin{aligned}
&\mathbb{E}\left[\widetilde{\frac{dF_{t_K}^{RT}}{dx}}\right] \\
=&\mathbb{E}\left[\widehat{\frac{dF_{t_1}}{dx}}\right] + \sum_{k=1}^{K} p_k \frac{1}{p_k} \mathbb{E}\left[\widetilde{\frac{dF_{t_{k+1}}}{dx}} - \widetilde{\frac{dF_{t_k}^{t_{k+1}}}{dx}}\right] \\
=&\mathbb{E}\left[\widehat{\frac{dF_{t_1}}{dx}}\right] + \sum_{k=1}^{K} p_k \frac{1}{p_k} \left(\mathbb{E}\left[\widehat{\frac{dF_{t_{k+1}}}{dx}}\right] - \mathbb{E}\left[\widehat{\frac{dF_{t_k}}{dx}}\right]\right) \\
=&\mathbb{E}\left[\widehat{\frac{dF_{t_K}}{dx}}\right].
\end{aligned}
$$

It follows directly that the hypergradient estimators obtained by RT-Q and vanilla soft Q-learning must have the same bias.

For a sampled $k \in \{1, \dots, K\}$, soft Q-learning has an iteration complexity c_k of $\mathcal{O}(k2^k)$ to build the following hypergradient estimator:

$$
\frac{dF_{t_K}^{RT}}{dx} = \widetilde{\frac{dF_{t_1}}{dx}} + \frac{\widetilde{\frac{dF_{t_{k+1}}}{dx}} - \widetilde{\frac{dF_{t_k}^{t_{k+1}}}{dx}}}{p_k}.
$$

As we sample any k with probability $p_k = \frac{2^{-k}}{1-2^{-K}}$, the expected iteration complexity is then given by

$$
\begin{aligned}
&\sum_{k=1}^{K} c_k p_k \\
=&\sum_{k=1}^{K} c_k \frac{2^{-k}}{1-2^{-K}} \\
=&\sum_{k=1}^{K} \mathcal{O}\left(\frac{k}{2^{-k}}\right) \frac{2^{-k}}{1-2^{-K}} \\
\leq&\mathcal{O}(K) \sum_{k=1}^{K} \mathcal{O}(\frac{1}{1-2^{-K}}) \\
=&\mathcal{O}(K^2),
\end{aligned}
$$

which proves our claim. The attentive reader will note that RT-Q runs `GradientEstimator` 2^k times instead of once and thus samples 2^k trajectories to estimate the advantage derivative instead of one like vanilla soft Q-learning. Nonetheless, RT-Q has the better sample complexity as both methods need to sample $\mathcal{O}(k2^k)$ state action pairs for a given k to run soft Q-learning and then estimate the advantage derivative. The same analysis as for the iteration complexity thus shows that the sample complexity of vanilla soft Q-learning is $\mathcal{O}(K2^K)$ and $\mathcal{O}(K^2)$ for RT-Q.

It remains to show that the variance is of order $\mathcal{O}(K)$. This is the most challenging part of the proof. As outlined previously, we will iteratively decompose the variance until we can bound all terms by $\mathcal{O}(K)$.

For better readability we introduce the following notation for a given pair s, a:

$$
H_k(1) = \frac{\partial_1 f(x, \pi^{t_k}, \xi)}{\partial x}
$$

$$H_k(2) = \begin{cases} \frac{1}{\lambda\nu(s)} \frac{\partial_2 f(x,\pi^{t_k},\xi)}{\partial\pi(s,a)} \widehat{\partial_x A^{\pi^{t_k}}}(s,a) & \text{if } k=1 \\ \frac{1}{\lambda\nu(s)} \frac{\partial_2 f(x,\pi^{t_k},\xi)}{\partial\pi(s,a)} \widetilde{\partial_x A^{\pi^{t_k}}}(s,a) & \text{if } k>1 \end{cases}$$

$$H_k^{k+1}(2) = \frac{1}{\lambda\nu(s)} \frac{\pi^{t_k}(a;s)}{\pi^{t_{k+1}}(a;s)} \frac{\partial_2 f(x,\pi^{t_k},\xi))}{\partial\pi(s,a)} \widetilde{\partial_x A^{\pi^{t_k}}}(s,a)$$

$$H^*(1) = \frac{\partial_1 f(x,\pi^*_{x,\xi},\xi)}{\partial x}$$

$$H^*(2) = \frac{1}{\lambda\nu(s)} \frac{\partial_2 f(x,\pi^*_{x,\xi},\xi))}{\partial\pi(s,a)} \partial_x A^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a).$$

Then for a given ξ, s, a (which are sampled at the beginning of RT-Q) we have that

$$\frac{d}{dx}F^{RT}_{t_K} = H_1(1) + H_1(2) + \frac{H_{k+1}(1) + H_{k+1}(2) - H_k(1) - H_k^{k+1}(2)}{p_k}.$$

Now let us decompose the variance of the RT-Q hypergradient estimator using the newly introduced notation:

$$\begin{aligned}
&\mathbb{E}\left[\left\|\frac{d}{dx}F^{RT}_{t_K} - \mathbb{E}\left[\frac{d}{dx}F^{RT}_{t_K}\right]\right\|^2_\infty\right] \\
\leq &\mathbb{E}\left[\left\|\frac{d}{dx}F^{RT}_{t_K} - \frac{d}{dx}F(x)\right\|^2_\infty\right] \\
\leq &2\mathbb{E}\left[\left\|\frac{d}{dx}F^{RT}_{t_K} - H_1(1) - H_1(2)\right\|^2_\infty\right] + 2\mathbb{E}\left[\left\|H_1(1) + H_1(2) - \frac{d}{dx}F(x)\right\|^2_\infty\right] \\
\leq &4\,\underbrace{\mathbb{E}\left[\left\|\frac{H_{k+1}(1) - H_k(1)}{p_k}\right\|^2_\infty\right]}_{\textbf{(1)}} + 4\,\underbrace{\mathbb{E}\left[\left\|\frac{H_{k+1}(2) - H_k^{k+1}(2)}{p_k}\right\|^2_\infty\right]}_{\textbf{(2)}} + 2\,\underbrace{\mathbb{E}\left[\left\|H_1(1) + H_1(2) - \frac{d}{dx}F(x)\right\|^2_\infty\right]}_{\textbf{(3)}}.
\end{aligned}$$

We proceed to bound the individual terms.

Note that **(3)** is indepedent of k. Using Equation (15) in the analysis of Theorem 3, we can bound it as

$$\begin{aligned}
\textbf{(3)} &\leq 4\left(\left(\frac{K_2}{1-\gamma} + \frac{K_1(\overline{R} + \lambda\log|\mathcal{A}|)}{(1-\gamma)^2}\right) + \frac{1}{1-\gamma}K_2 + \frac{\gamma}{1-\gamma}K_1\frac{\overline{R} + \lambda\log|\mathcal{A}|}{1-\gamma^{0.5}}\right)^2 \\
&= \mathcal{O}(1).
\end{aligned}$$

(1) is also relatively easy to bound, as shown below:

$$\begin{aligned}
\textbf{(1)} &= \sum_{k=1}^K \frac{1}{p_k}\mathbb{E}\left[\|H_{k+1}(1) - H_k(1)\|^2_\infty\right] \\
&= \sum_{k=1}^K \frac{1}{p_k}\mathbb{E}\left[\left\|\frac{\partial_1 f(x,\pi^{t_{k+1}},\xi)}{\partial x} - \frac{\partial_1 f(x,\pi^{t_k},\xi)}{\partial x}\right\|^2_\infty\right] \\
&\leq \sum_{k=1}^K \frac{1}{p_k}S_f^2\mathbb{E}\left[\left\|\pi^{t_{k+1}} - \pi^{t_k}\right\|^2_\infty\right] \\
&\leq \sum_{k=1}^K \frac{1}{p_k}S_f^2 2\left(\mathbb{E}\left[\left\|\pi^{t_{k+1}} - \pi^*_{x,\xi}\right\|^2_\infty\right] + \mathbb{E}\left[\left\|\pi^{t_k} - \pi^*_{x,\xi}\right\|^2_\infty\right]\right) \\
&\leq \sum_{k=1}^K \frac{1}{p_k}S_f^2 2\left(\frac{1}{2^{k+1}} + \frac{1}{2^k}\right)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{k=1}^{K} 2^k S_f^2 \frac{3}{2^k} \\
&= 3 S_f^2 K \\
&= \mathcal{O}(K).
\end{aligned}
$$

In the first equality we simply use that k is sampled with probability $p_k = \frac{2^{-k}}{1-2^{-K}}$. In the second equality we plug in the definitions of $H_{k+1}(1)$ and $H_k(1)$. In the first inequality we use the smoothness of f. The second inequality uses Equation (16). The third inequality follows from the fact that t_{k+1} is chosen to guarantee an expected distance of at most $\frac{1}{2^{k+1}}$ to $\pi^*_{x,\xi}$. The remaining equalities follow from plugging in p_k and rearranging terms.

Now we want to repeat the same analysis again for (2). As for (1) we get a sum over k with the factor $\frac{1}{p_k}$, which we need to compensate by bounding $\mathbb{E}\left[\left\|H_{k+1}(2) - H_k^{k+1}(2)\right\|_\infty^2\right]$ by $\mathcal{O}(2^{-k})$. However, bounding the latter is more invovled than our analysis for (1). In the following pages, we will iteratively apply Equation (16) until the terms get easy enough, such that we can use one of the following two facts about RT-Q:

1. $t_k = \mathcal{O}(k2^k)$ is chosen sucht that $\lambda \mathbb{E}\|\pi^{t_k} - \pi^*\|_\infty^2 \leq \mathbb{E}\|Q^{t_k} - Q^*_\lambda\|_\infty^2 \leq 2^{-k}$
2. $\widetilde{\partial_x A^{\pi^{t_{k+1}}}}(s,a)$ and related terms are an average estimate over 2^k trajectory samples, such that their variance (with respect to these random rollouts) is $\mathcal{O}(\frac{1}{2^k})$.

We briefly note that in the analysis below we will sometimes write $Q(s,a)$ or $A(s,a)$ inside the infinity norm for ease of exposition. When we do so, the infinity norm is still interpreted as the maximum over all possible s, a and not as the absolute value of the Q-function or advantage for a specific s, a.

Let us start decomposing the numerator of (2):

$$
\begin{aligned}
&\mathbb{E}\left[\left\|H_{k+1}(2) - H_k^{k+1}(2)\right\|_\infty^2\right] \\
=&\mathbb{E}\left[\left\|\frac{1}{\lambda\nu(s)}\frac{\partial_2 f(x,\pi^{t_k},\xi))}{\partial\pi(s,a)}\widetilde{\partial_x A^{\pi^{t_k}}}(s,a) - \frac{1}{\lambda\nu(s)}\frac{\pi^{t_k}(a;s)}{\pi^{t_{k+1}}(a;s)}\frac{\partial_2 f(x,\pi^{t_k},\xi))}{\partial\pi(s,a)}\widetilde{\partial_x A^{\pi^{t_k}}}(s,a)\right\|_\infty^2\right] \\
\leq&2\mathbb{E}\left[\left\|\frac{1}{\lambda\nu(s)}\frac{\partial_2 f(x,\pi^{t_k},\xi))}{\partial\pi(s,a)}\widetilde{\partial_x A^{\pi^{t_k}}}(s,a)\right\|_\infty^2 \left\|\frac{\pi^{t_{k+1}}(a;s)}{\pi^{t_{k+1}}(a;s)} - \frac{\pi^{t_k}(a;s)}{\pi^{t_{k+1}}(a;s)}\right\|_\infty^2\right] \\
&+ 2\mathbb{E}\left[\left\|\frac{\pi^{t_k}(a;s)}{\pi^{t_{k+1}}(a;s)}\right\|_\infty^2 \left\|\frac{1}{\lambda\nu(s)}\frac{\partial_2 f(x,\pi^{t_{k+1}},\xi))}{\partial\pi(s,a)}\widetilde{\partial_x A^{\pi^{t_{k+1}}}}(s,a) - \frac{1}{\lambda\nu(s)}\frac{\partial_2 f(x,\pi^{t_k},\xi))}{\partial\pi(s,a)}\widetilde{\partial_x A^{\pi^{t_k}}}(s,a)\right\|_\infty^2\right] \\
\leq&\underbrace{2\,\mathbb{E}\left[\left\|\frac{1}{\lambda\nu(s)}\frac{\partial_2 f(x,\pi^{t_k},\xi))}{\partial\pi(s,a)}\widetilde{\partial_x A^{\pi^{t_k}}}(s,a)\right\|_\infty^2 \left\|\pi^{t_{k+1}}(a;s) - \pi^{t_k}(a;s)\right\|_\infty^2 \left\|\frac{1}{\pi^{t_{k+1}}(a;s)}\right\|_\infty^2\right]}_{\text{(i)}} \\
&\underbrace{+\,2\,\mathbb{E}\left[\left\|\frac{\pi^{t_k}(a;s)}{\pi^{t_{k+1}}(a;s)}\right\|_\infty^2 \left\|\frac{1}{\lambda\nu(s)}\frac{\partial_2 f(x,\pi^{t_{k+1}},\xi))}{\partial\pi(s,a)}\widetilde{\partial_x A^{\pi^{t_{k+1}}}}(s,a) - \frac{1}{\lambda\nu(s)}\frac{\partial_2 f(x,\pi^{t_k},\xi))}{\partial\pi(s,a)}\widetilde{\partial_x A^{\pi^{t_k}}}(s,a)\right\|_\infty^2\right]}_{\text{(ii)}}
\end{aligned}
$$

The first equality is just plugging in definitions. The first inequality follows from Equation (16) and the second inequality comes from the Cauchy-Schwarz inequality.

Let us begin with bounding (i). First we want to tackle the fractions of the policies. For this, recall that $\forall x, \xi, s, a : |r_{x,\xi}(s,a)| < \overline{R}$ and that $0 \leq H(\pi; s) \leq \log|\mathcal{A}|$ (by Jensen's inequality). Thus we have that:

$$
\frac{-\overline{R}}{1-\gamma} \leq Q_\lambda(s,a) \leq \frac{\overline{R} + \lambda \log|\mathcal{A}|}{1-\gamma}.
$$

The above bounds extend to any soft Q-learning estimate, which can be obtained by Algorithm 4, since by Assumption 3.1 the algorithm cannot observe rewards with greater magnitude than $\overline{R}$. As Algorithm 4 returns a softmax policy, i.e.

$$\pi^{t_k}(a;s) = \frac{\exp(Q^{t_k}(a;s)/\lambda)}{\sum_{a'}\exp(Q^{t_k}(a';s)/\lambda)},$$

it holds that:

$$\forall t_k, \forall s, \forall a : \frac{\exp(\frac{\overline{R}+\lambda\log|\mathcal{A}|}{\lambda(1-\gamma)})}{|\mathcal{A}|\exp(\frac{-\overline{R}}{\lambda(1-\gamma)})} \geq \pi^{t_k}(a;s) \geq \frac{\exp(\frac{-\overline{R}}{\lambda(1-\gamma)})}{|\mathcal{A}|\exp(\frac{\overline{R}+\lambda\log|\mathcal{A}|}{\lambda(1-\gamma)})}. \tag{17}$$

Therefore we have:

$$\exists M_1 < \infty, \forall t_k : \left\|\frac{1}{\pi^{t_k}(a;s)}\right\|_\infty^2 \leq M_1 -$$
$$\exists M_2 < \infty, \forall t_k : \left\|\frac{\pi^{t_k}(a;s)}{\pi^{t_{k+1}}(a;s)}\right\|_\infty^2 \leq M_2.$$

Using this we can bound (i). Let $m = \min_s \nu(s)$, then:

$$\begin{aligned}
\text{(i)} &\leq \frac{M_1}{\lambda m} 2\left(\frac{K_2}{1-\gamma} + \frac{\gamma}{1-\gamma}K_1\frac{\overline{R}+\lambda\log|\mathcal{A}|}{1-\gamma^{0.5}}\right)\mathbb{E}\left[\left\|\pi^{t_{k+1}}(a;s) - \pi^{t_k}(a;s)\right\|_\infty^2\right] \\
&\leq \mathcal{O}\left(2\mathbb{E}\left[\left\|\pi^{t_{k+1}}(a;s) - \pi^*_{x,\xi}(a;s)\right\|_\infty^2\right] + 2\mathbb{E}\left[\left\|\pi^{t_k}(a;s) - \pi^*_{x,\xi}(a;s)\right\|_\infty^2\right]\right) \\
&\leq \mathcal{O}\left(\frac{1}{2^k}\right).
\end{aligned}$$

The first inequality uses Equation (15). The second inequality uses Equation (16) and the last equality uses the convergence of soft Q-learning.

Now we turn to bound **(ii)**:

$$\begin{aligned}
\text{(ii)} \leq& 2\frac{M_2}{\lambda m}\mathbb{E}\left[\left\|\frac{\partial_2 f(x,\pi^{t_{k+1}},\xi))}{\partial\pi(s,a)}\widetilde{\partial_x A^{\pi^{t_{k+1}}}}(s,a) - \frac{\partial_2 f(x,\pi^{t_k},\xi))}{\partial\pi(s,a)}\widetilde{\partial_x A^{\pi^{t_k}}}(s,a)\right\|_\infty^2\right] \\
\leq& 2\frac{M_2}{\lambda m}\mathbb{E}\Bigg[2\left\|\frac{\partial_2 f(x,\pi^{t_{k+1}},\xi))}{\partial\pi(s,a)}\widetilde{\partial_x A^{\pi^{t_{k+1}}}}(s,a) - \frac{\partial_2 f(x,\pi^{t_{k+1}},\xi))}{\partial\pi(s,a)}\widetilde{\partial_x A^{\pi^{t_k}}}(s,a)\right\|_\infty^2 \\
&+ 2\left\|\frac{\partial_2 f(x,\pi^{t_{k+1}},\xi))}{\partial\pi(s,a)}\widetilde{\partial_x A^{\pi^t_k}}(s,a) - \frac{\partial_2 f(x,\pi^{t_k},\xi))}{\partial\pi(s,a)}\widetilde{\partial_x A^{\pi^{t_k}}}(s,a)\right\|_\infty^2\Bigg] \\
\leq& 4\frac{M_2}{\lambda m}\Bigg(\mathbb{E}\left[\left\|\frac{\partial_2 f(x,\pi^{t_{k+1}},\xi))}{\partial\pi(s,a)}\right\|_\infty^2\left\|\widetilde{\partial_x A^{\pi^{t_{k+1}}}}(s,a) - \widetilde{\partial_x A^{\pi^{t_k}}}(s,a)\right\|_\infty^2\right. \\
&\left.+ \left\|\widetilde{\partial_x A^{\pi^{t_k}}}(s,a)\right\|_\infty^2\left\|\frac{\partial_2 f(x,\pi^{t_{k+1}},\xi))}{\partial\pi(s,a)} - \frac{\partial_2 f(x,\pi^{t_k},\xi))}{\partial\pi(s,a)}\right\|_\infty^2\right]\Bigg) \\
\leq& 4\frac{M_2}{\lambda m}\Bigg(\underbrace{\mathbb{E}\left[\left\|\widetilde{\partial_x A^{\pi^{t_k}}}(s,a)\right\|_\infty^2\left\|\frac{\partial_2 f(x,\pi^{t_{k+1}},\xi))}{\partial\pi(s,a)} - \frac{\partial_2 f(x,\pi^{t_k},\xi))}{\partial\pi(s,a)}\right\|_\infty^2\right]}_{\text{(A)}} \\
&+ L_f^2\underbrace{\mathbb{E}\left[\left\|\widetilde{\partial_x A^{\pi^{t_{k+1}}}}(s,a) - \widetilde{\partial_x A^{\pi^{t_k}}}(s,a)\right\|_\infty^2\right]}_{\text{(B)}}\Bigg).
\end{aligned}$$

The first inequality uses M_2 to bound the policy fraction the definition of $m = \min_s \nu(s)$ and Cauchy-Schwarz. The second inequality uses Equation (16) and Cauchy-Schwarz. For the final inequality, we rearrange and use the Lipschitz continuity of f (cf. Assumption 3.1).

(A) is relatively easy to bound as follows:

$$\begin{aligned}
\text{(A)} &\leq \left(2\left(\frac{K_2}{1-\gamma}+\frac{\gamma}{1-\gamma}K_1\frac{\overline{R}+\lambda\log|\mathcal{A}|}{1-\gamma^{0.5}}\right)\right)^2 S_f\mathbb{E}\left[\left\|\pi^{t_{k+1}}(a;s)-\pi^{t_k}(a;s)\right\|_\infty^2\right] \\
&\leq \mathcal{O}\left(2\mathbb{E}\left[\left\|\pi^{t_{k+1}}(a;s)-\pi^*_{x,\xi}(a;s)\right\|_\infty^2\right]+2\mathbb{E}\left[\left\|\pi^{t_k}(a;s)-\pi^*_{x,\xi}(a;s)\right\|_\infty^2\right]\right) \\
&\leq \mathcal{O}\left(\frac{1}{2^k}\right).
\end{aligned}$$

In the first inequality we use the smoothness of f and Equation (15). In the second inequality we use Equation (16) and in the final inequality the convergence of soft Q-learning.

Now, we turn our attention towards bounding (B). First, notice that both advantage derivatives are evaluated for the same state-action pair. This is because in Algorithm 8 we sample a once according to $\pi^{t_{k+1}}$ and reuse the same a for $A^{\pi^{t_k}}(s,a)$ by doing importance sampling. Without this trick, it would be hopeless to bound (B).

When bounding (B), we need to consider two sources of randomness. First, there is the randomness in the soft Q-learning iterations. Second, we have to account for the randomness over the trajectory rollouts of `GradientEstimator` to estimate $\widetilde{\partial_x A^\pi}$. `GradienEstimator` first separately estimates $\widehat{\partial_x V^{\pi^{t_k}}}$ and $\widehat{\partial_x V^{\pi^{t_k}}}$ from a trajectory and then returns $\widehat{\partial_x A^{\pi^{t_k}}} = \widehat{\partial_x Q^{\pi^{t_k}}} - \widehat{\partial_x V^{\pi^{t_k}}}$. We therefore denote by

$$\widetilde{\partial_x V^{\pi^{t_k}}} := \frac{1}{2^k}\sum_{l=1}^{2^k}\widehat{\partial_x V^{\pi^{t_k}}}(\tau_l),$$

the average over the 2^k value function derivatives estimated as part of the 2^k `GradientEstimator` procedures performed in RT-Q. Note because of the unbiasedness of `GradienEstimator` (cf. Proposition 3), it holds that $\partial_x V^{\pi^{t_k}} = \mathbb{E}_\tau\left[\widetilde{\partial_x V^{\pi^{t_k}}}\right]$

Using the above notation, we get the following bound by repeatedly applying Equation (16):

$$\begin{aligned}
\text{(B)} \leq& 2\mathbb{E}\left\|\widetilde{\partial_x Q^{\pi^{t_k}}}(s,a)-\widetilde{\partial_x Q^{\pi^{t_{k+1}}}}(s,a)\right\|_\infty^2+2\mathbb{E}\left\|\widetilde{\partial_x V^{\pi^{t_k}}}(s)-\widetilde{\partial_x V^{\pi^{t_{k+1}}}}(s)\right\|_\infty^2 \\
\leq& 4\mathbb{E}\left\|\widetilde{\partial_x Q^{\pi^{t_k}}}(s,a)-\partial_x Q^{\pi^*}(s,a)\right\|_\infty^2+4\mathbb{E}\left\|\widetilde{\partial_x Q^{\pi^{t_{k+1}}}}(s,a)-\partial_x Q^{\pi^*}(s,a)\right\|_\infty^2 \\
&+4\mathbb{E}\left\|\widetilde{\partial_x V^{\pi^{t_k}}}(s)-\partial_x V^{\pi^*}(s)\right\|_\infty^2+4\mathbb{E}\left\|\widetilde{\partial_x V^{\pi^{t_{k+1}}}}(s)-\partial_x V^{\pi^*}(s)\right\|_\infty^2 \\
\leq& 8\mathbb{E}\left\|\widetilde{\partial_x Q^{\pi^{t_k}}}(s,a)-\partial_x Q^{\pi^{t_k}}(s,a)\right\|_\infty^2+8\mathbb{E}\left\|\partial_x Q^{\pi^{t_k}}(s,a)-\partial_x Q^{\pi^*}(s,a)\right\|_\infty^2 \\
&+8\mathbb{E}\left\|\widetilde{\partial_x Q^{\pi^{t_{k+1}}}}(s,a)-\partial_x Q^{\pi^{t_{k+1}}}(s,a)\right\|_\infty^2+8\mathbb{E}\left\|\partial_x Q^{\pi^{t_{k+1}}}(s,a)-\partial_x Q^{\pi^*}(s,a)\right\|_\infty^2 \\
&+8\mathbb{E}\left\|\widetilde{\partial_x V^{\pi^{t_k}}}(s)-\partial_x V^{\pi^{t_k}}(s)\right\|_\infty^2+8\mathbb{E}\left\|\partial_x V^{\pi^{t_k}}(s)-\partial_x V^{\pi^*}(s)\right\|_\infty^2 \\
&+8\mathbb{E}\left\|\widetilde{\partial_x V^{\pi^{t_{k+1}}}}(s)-\partial_x V^{\pi^{t_{k+1}}}(s)\right\|_\infty^2+8\mathbb{E}\left\|\partial_x V^{\pi^{t_{k+1}}}(s)-\partial_x V^{\pi^*}(s)\right\|_\infty^2.
\end{aligned}$$

Here Q^*_λ denotes the optimal Q-function, i.e. for the policy $\pi^*_{x,\xi}$. In the equations above we have two flavours of terms. The first are the differences between the derivative estimator and its expectation for a given policy and the second are the differences between the expected derivative under the learned and optimal policy. We start by bounding the first kind of terms. Recall that

$$\widetilde{\partial_x Q^{\pi^{t_k}}} = \frac{1}{2^k}\sum_{l=1}^{2^k}\widehat{\partial_x Q^{\pi^{t_k}}}(\tau_l), \quad \widetilde{\partial_x V^{\pi^{t_k}}} = \frac{1}{2^k}\sum_{l=1}^{2^k}\widehat{\partial_x V^{\pi^{t_k}}}(\tau_l)$$

As the second moment of $\widehat{\partial_x Q^{\pi^{t_k}}}(\tau_l)$ and $\widehat{\partial_x V^{\pi^{t_k}}}(\tau_l)$ is bounded, we have that:

$$\mathbb{E}_\tau\left[\left\|\widetilde{\partial_x Q^{\pi^{t_k}}}-\partial_x Q^{\pi^{t_k}}\right\|_\infty^2\right]=\mathcal{O}(2^{-k}),\quad \mathbb{E}_\tau\left[\left\|\widetilde{\partial_x V^{\pi^{t_k}}}-\partial_x V^{\pi^{t_k}}\right\|_\infty^2\right]=\mathcal{O}(2^{-k}) \tag{18}$$

since we use 2^k sampled trajectories. The same analysis of course also holds for t_{k+1} and thus plugging this in, we get

$$\begin{aligned}
\textbf{(B)} \leq& \mathcal{O}(\frac{1}{2^k})+8\mathbb{E}\left\|\partial_x Q^{\pi^{t_k}}(s,a)-\partial_x Q^{\pi^*}(s,a)\right\|_\infty^2+8\mathbb{E}\left\|\partial_x Q^{\pi^{t_{k+1}}}(s,a)-\partial_x Q^{\pi^*}(s,a)\right\|_\infty^2\\
&+8\mathbb{E}\left\|\partial_x V^{\pi^{t_k}}(s)-\partial_x V^{\pi^*}(s)\right\|_\infty^2+8\mathbb{E}\left\|\partial_x V^{\pi^{t_{k+1}}}(s)-\partial_x V^{\pi^*}(s)\right\|_\infty^2\\
\leq& \mathcal{O}(\frac{1}{2^k})+24\mathbb{E}\left\|\partial_x Q^{\pi^{t_k}}(s,a)-\partial_x Q^{\pi^*}(s,a)\right\|_\infty^2+24\mathbb{E}\left\|\partial_x Q^{\pi^{t_{k+1}}}(s,a)-\partial_x Q^{\pi^*}(s,a)\right\|_\infty^2\\
&+16|\mathcal{A}|^2\left(\frac{K_1(\overline{R}+\lambda\log|\mathcal{A}|)}{(1-\gamma)^2}\right)^2\left(\mathbb{E}\left\|\pi^{t_k}-\pi^*\right\|_\infty^2+\mathbb{E}\left\|\pi^{t_{k+1}}-\pi^*\right\|_\infty^2\right)\\
\leq& \mathcal{O}(\frac{1}{2^k})+24\mathbb{E}\left\|\partial_x Q^{\pi^{t_k}}(s,a)-\partial_x Q^{\pi^*}(s,a)\right\|_\infty^2+24\mathbb{E}\left\|\partial_x Q^{\pi^{t_{k+1}}}(s,a)-\partial_x Q^{\pi^*}(s,a)\right\|_\infty^2.
\end{aligned}$$

In the first inequality, we simply plug in Equation (18). In the third inequality we use the convergence of soft Q-learning and in the second inequality, we use the following identity (cf. Equation (14)):

$$\begin{aligned}
&\left\|\partial_x V_{\lambda,x,\xi}^{\pi^o_{x_t,\xi}}(s)-\partial_x V_{\lambda,x,\xi}^{\pi^*_{x_t,\xi}}(s)\right\|_\infty^2\\
=&\left\|\sum_a \pi^o_{x_t,\xi}(a;s)\partial_x Q_{\lambda,x,\xi}^{\pi^o_{x_t,\xi}}(s,a)-\sum_a \pi^*_{x_t,\xi}(a;s)\partial_x Q_{\lambda,x,\xi}^{\pi^*_{x_t,\xi}}(s,a)\right\|_\infty^2\\
=&\left\|\sum_a\left(\pi^o_{x_t,\xi}(a;s)-\pi^*_{x_t,\xi}(a;s)\right)\partial_x Q_{\lambda,x,\xi}^{\pi^o_{x_t,\xi}}(s,a)-\sum_a \pi^*_{x_t,\xi}(a;s)\left(\partial_x Q_{\lambda,x,\xi}^{\pi^*_{x_t,\xi}}(s,a)-\partial_x Q_{\lambda,x,\xi}^{\pi^o_{x_t,\xi}}(s,a)\right)\right\|_\infty^2\\
\leq&2|\mathcal{A}|^2\left(\frac{K_1(\overline{R}+\lambda\log|\mathcal{A}|)}{(1-\gamma)^2}\right)^2\left\|\pi^o_{x_t,\xi}-\pi^*_{x_t,\xi}\right\|_\infty+2\left\|\partial_x Q_{\lambda,x,\xi}^{\pi^*_{x_t,\xi}}(s,a)-\partial_x Q_{\lambda,x,\xi}^{\pi^o_{x_t,\xi}}(s,a)\right\|_\infty^2,
\end{aligned}$$

where we use Equation (16) for the last inequality.

To bound **(B)**, it only remains to upper bound $\mathbb{E}\left\|\partial_x Q^{\pi^{t_k}}(s,a)-\partial_x Q^{\pi^*}(s,a)\right\|_\infty^2$ and $\mathbb{E}\left\|\partial_x Q^{\pi^{t_{k+1}}}(s,a)-\partial_x Q^{\pi^*}(s,a)\right\|_\infty^2$. Below we will derive an upper bound for the former term. The same analysis yields an equivalent bound for the latter term. We denote by V_λ^* the optimal regularized value function. Note, the analysis is similar to the one performed in the proof of Theorem 3,

$$\begin{aligned}
&\left\|\partial_x Q^{\pi^{t_k}}(s,a)-\partial_x Q_\lambda^*(s,a)\right\|_\infty\\
=&\left\|\sum_{t=0}^\infty\sum_{s',a'}\gamma^t p_{x,\xi}(s,a\to s',a';t,\pi^{t_k})\left(\frac{dr_{x,\xi}(s',a')}{dx}+\gamma\sum_{s''}\frac{dP_{x,\xi}(s'';s',a')}{dx}V^{\pi^{t_k}}(s'')\right)\right.\\
&\left.-\sum_{t=0}^\infty\sum_{s',a'}\gamma^t p_{x,\xi}(s,a\to s',a';t,\pi^*_{x_t,\xi})\left(\frac{dr_{x,\xi}(s',a')}{dx}+\gamma\sum_{s''}\frac{dP_{x,\xi}(s'';s',a')}{dx}V_\lambda^*(s'')\right)\right\|_\infty\\
=&\left\|\frac{dr_{x,\xi}(s,a)}{dx}+\gamma\sum_{s'}\frac{dP_{x,\xi}(s';s,a)}{dx}V^{\pi^{t_k}}(s')\right.\\
&+\sum_{t=1}^\infty\sum_{s',a'}\gamma^t p_{x,\xi}(s,a\to s',a';t,\pi^{t_k})\left(\frac{dr_{x,\xi}(s',a')}{dx}+\gamma\sum_{s''}\frac{dP_{x,\xi}(s'';s',a')}{dx}V^{\pi^{t_k}}(s'')\right)\\
&\left.-\frac{dr_{x,\xi}(s,a)}{dx}-\gamma\sum_{s'}\frac{dP_{x,\xi}(s';s,a)}{dx}V_\lambda^*(s')\right.
\end{aligned}$$

$$
\begin{aligned}
&- \sum_{t=1}^{\infty} \sum_{s',a'} \gamma^t p_{x,\xi}(s, a \to s', a'; t, \pi^*_{x_t,\xi}) \left(\frac{dr_{x,\xi}(s',a')}{dx} + \gamma \sum_{s''} \frac{dP_{x,\xi}(s'';s',a')}{dx} V^*_\lambda(s'') \right) \Bigg\|_\infty \\
\leq & \gamma |\mathcal{S}| \left\| \frac{dP_{x,\xi}(s';s,a)}{dx} \right\|_\infty \left\| V^{\pi^{t_k}}(s') - V^*_\lambda(s') \right\|_\infty \\
&+ \gamma \Bigg\| \sum_{s',a'} P(s';s,a) \pi^*_{x_t,\xi}(a',s') \sum_{t=0}^{\infty} \sum_{s'',a''} \gamma^t p_{x,\xi}(s', a' \to s'', a''; t, \pi^*_{x_t,\xi}) \dots \\
&\quad \left(\frac{dr_{x,\xi}(s'',a'')}{dx} + \gamma \sum_{s'''} \frac{dP_{x,\xi}(s''';s'',a'')}{dx} V^*_\lambda(s'') \right) \\
&- \sum_{s',a'} P(s';s,a) \pi^{t_k}(a',s') \sum_{t=0}^{\infty} \sum_{s'',a''} \gamma^t p_{x,\xi}(s', a' \to s'', a''; t, \pi^{t_k}) \dots \\
&\quad \left(\frac{dr_{x,\xi}(s'',a'')}{dx} + \gamma \sum_{s'''} \frac{dP_{x,\xi}(s''';s'',a'')}{dx} V^{\pi^{t_k}}(s'') \right) \Bigg\|_\infty \\
\leq & \gamma |\mathcal{S}| \left\| \frac{dP_{x,\xi}(s';s,a)}{dx} \right\|_\infty \left\| V^{\pi^{t_k}}(s') - V^*_\lambda(s') \right\|_\infty \\
&+ \gamma \Bigg\| \sum_{s',a'} P(s';s,a) \pi^*_{x_t,\xi}(a',s') \partial_x Q^*_\lambda(s',a') \\
&- \sum_{s',a'} P(s';s,a) \pi^{t_k}(a',s') \partial_x Q^{\pi^{t_k}}(s',a') \Bigg\|_\infty \\
\leq & \gamma |\mathcal{S}| \left\| \frac{dP_{x,\xi}(s';s,a)}{dx} \right\|_\infty \left\| V^{\pi^{t_k}}(s') - V^*_\lambda(s') \right\|_\infty \\
&+ \gamma \left\| \partial_x Q^*_\lambda(s',a') \right\|_\infty \left\| \sum_{s'} P(s';s,a) \right\|_\infty \left\| \pi^*_{x_t,\xi} - \pi^{t_k} \right\|_\infty \\
&+ \gamma \sum_{s',a'} P(s';s,a) \pi^{t_k}(a',s') \left\| \partial_x Q^*_\lambda(s',a') - \partial_x Q^{\pi^{t_k}}(s',a') \right\|_\infty \\
\leq & \gamma |\mathcal{S}| \left\| \frac{dP_{x,\xi}(s';s,a)}{dx} \right\|_\infty \left\| V^{\pi^{t_k}}(s') - V^*_\lambda(s') \right\|_\infty \\
&+ \gamma \left\| \partial_x Q^*_\lambda(s',a') \right\|_\infty \left\| \pi^*_{x_t,\xi} - \pi^{t_k} \right\|_\infty \\
&+ \gamma \left\| \partial_x Q^*_\lambda(s',a') - \partial_x Q^{\pi^{t_k}}(s',a') \right\|_\infty \\
\leq & \gamma |\mathcal{S}| K_1 \left\| V^{\pi^{t_k}}(s') - V^*_\lambda(s') \right\|_\infty \\
&+ \gamma \left(\frac{K_2}{1-\gamma} + \frac{K_1(\overline{R} + \lambda \log \mathcal{A})}{(1-\gamma)^2} \right) \left\| \pi^*_{x_t,\xi} - \pi^{t_k} \right\|_\infty \\
&+ \gamma \left\| \partial_x Q^*_\lambda(s',a') - \partial_x Q^{\pi^{t_k}}(s',a') \right\|_\infty \\
\leq & \frac{\gamma}{1-\gamma} |\mathcal{S}| K_1 \left\| V^{\pi^{t_k}}(s') - V^*_\lambda(s') \right\|_\infty \\
&+ \frac{\gamma}{1-\gamma} \left(\frac{K_2}{1-\gamma} + \frac{K_1(\overline{R} + \lambda \log \mathcal{A})}{(1-\gamma)^2} \right) \left\| \pi^*_{x_t,\xi} - \pi^{t_k} \right\|_\infty
\end{aligned}
$$

The first equality follows from plugging in the result from Theorem 2. The second equality follows by taking out all terms with $t = 0$. The first inequality used the triangle inequality. The second inequality plugs back in the definition from Theorem 2. The third inequality uses Cauchy-Schwarz.

The fourth inequality follows from simplifying. The fifth inequality uses Equation (12). The sixth inequality uses the geometric sum.

To simplify the bound above we want to expres $\left\|V^{\pi^{t_k}}(s') - V^*_\lambda(s')\right\|_\infty$ using $\left\|Q^{\pi^{t_k}} - Q^*_\lambda\right\|_\infty$ and $\left\|\pi^{t_k} - \pi^*\right\|_\infty$ (as we know the latter two converge for soft Q-learning). We have previously seen in Equation (13) that

$$\left\|V^{\pi^{t_k}}(s') - V^*_\lambda(s')\right\|_\infty \leq \left\|Q^{\pi^{t_k}} - Q^*_\lambda\right\|_\infty + \left\|\pi^{t_k} - \pi^*\right\|_\infty \left(\frac{\overline{R}}{1-\gamma} + \lambda|\mathcal{A}||\log l_2| + \frac{2}{l_2}\right).$$

Here we use l_2 to denote the minimum possible value that any policy output by soft Q-learning can achieve. Note $l_2 > 0$, which follows from Equation (17).

Plugging this result back in, we get

$$\begin{aligned}
&\left\|\partial_x Q^{\pi^{t_k}}(s,a) - \partial_x Q^*_\lambda(s,a)\right\|_\infty \\
\leq &\frac{\gamma}{1-\gamma}|\mathcal{S}|K_1\left(\left\|Q^{\pi^{t_k}} - Q^*_\lambda\right\|_\infty + \left\|\pi^{t_k} - \pi^*\right\|_\infty \left(\frac{\overline{R}}{1-\gamma} + \lambda|\mathcal{A}||\log l_2| + \frac{2}{l_2}\right)\right) \\
&+ \frac{\gamma}{1-\gamma}\left(\frac{K_2}{1-\gamma} + \frac{K_1(\overline{R} + \lambda\log\mathcal{A})}{(1-\gamma)^2}\right)\left\|\pi^*_{x_t,\xi} - \pi^{t_k}\right\|_\infty
\end{aligned}$$

From [49][Lemma 24] we know

$$\left\|\pi^{t_k} - \pi^*\right\|_\infty \leq \frac{1}{\lambda}\left\|Q^{t_k} - Q^*_\lambda\right\|_\infty.$$

With that result we can simplify to

$$\begin{aligned}
&\left\|\partial_x Q^{\pi^{t_k}}(s,a) - \partial_x Q^*_\lambda(s,a)\right\|_\infty \\
\leq &\frac{\gamma}{1-\gamma}|\mathcal{S}|K_1\left(\left\|Q^{\pi^{t_k}} - Q^*_\lambda\right\|_\infty + \frac{1}{\lambda}\left\|Q^{t_k} - Q^*_\lambda\right\|_\infty \left(\frac{\overline{R}}{1-\gamma} + \lambda|\mathcal{A}||\log l_2| + \frac{2}{l_2}\right)\right) \\
&+ \frac{\gamma}{1-\gamma}\left(\frac{K_2}{1-\gamma} + \frac{K_1(\overline{R} + \lambda\log\mathcal{A})}{(1-\gamma)^2}\right)\frac{1}{\lambda}\left\|Q^{t_k} - Q^*_\lambda\right\|_\infty \\
\leq &\left\|Q^{t_k} - Q^*_\lambda\right\|_\infty \left(\frac{\gamma}{1-\gamma}|\mathcal{S}|K_1\left(1 + \frac{1}{\lambda}\frac{\overline{R}}{1-\gamma} + |\mathcal{A}||\log l_2| + \frac{2}{\lambda l_2}\right) + \frac{\gamma}{1-\gamma}\left(\frac{K_2}{1-\gamma} + \frac{K_1(\overline{R} + \lambda\log\mathcal{A})}{(1-\gamma)^2}\right)\frac{1}{\lambda}\right).
\end{aligned}$$

Therefore it follows that

$$\mathbb{E}\left[\left\|\partial_x Q^{\pi^{t_k}}(s,a) - \partial_x Q^*_\lambda(s,a)\right\|^2_\infty\right] = \mathcal{O}\left(\frac{1}{2^k}\right),$$

where we use the convergence of soft Q-learning.

Using the above analysis, we can now bound:

$$\begin{aligned}
\textbf{(B)} &\leq \mathcal{O}(\frac{1}{2^k}) + 24\mathbb{E}\left\|\partial_x Q^{\pi^{t_k}}(s,a) - \partial_x Q^{\pi^*}(s,a)\right\|^2_\infty + 24\mathbb{E}\left\|\partial_x Q^{\pi^{t_{k+1}}}(s,a) - \partial_x Q^{\pi^*}(s,a)\right\|^2_\infty \\
&= \mathcal{O}\left(\frac{1}{2^k}\right).
\end{aligned}$$

This allows us to finish bounding **(ii)** as follows:

$$\begin{aligned}
\textbf{(ii)} &\leq 4\frac{M_2}{\lambda m}\textbf{(A)} + L_f^2\textbf{(B)} \\
&= \mathcal{O}(\frac{1}{2^k}).
\end{aligned}$$

Plugging this result into (2), we thus have:

$$\begin{aligned}
\textbf{(2)} &\leq \sum_{k=1}^{K} \frac{2}{p_k} \left(\textbf{(i)} + \textbf{(ii)}\right) \\
&= \sum_{k=1}^{K} 2^k \left(\mathcal{O}(\frac{1}{2^k}) + \mathcal{O}(\frac{1}{2^k})\right) \\
&= \mathcal{O}(K).
\end{aligned}$$

Coming back to the start, we get for the variance our desired result as follows:

$$\begin{aligned}
&\mathbb{E}\left[\left\| \frac{d}{dx} F_{t_K}^{RT} - \mathbb{E}\left[\frac{d}{dx} F_{t_K}^{RT}\right]\right\|_{\infty}^{2}\right] \\
&\leq 4\textbf{(1)} + 4\textbf{(2)} + 2\textbf{(3)} \\
&= \mathcal{O}(K) + \mathcal{O}(K) + \mathcal{O}(1) \\
&= \mathcal{O}(K).
\end{aligned}$$

□

E.6 Proof of Proposition 1

Proof. In this proof we will derive an expression for

$$\frac{df(x, \pi_{\lambda,x}^{*}, \xi)}{dx}.$$

Applying the Dominated Convergence Theorem then directly gives the expression for the derivative of $F(x)$. As we focus on f we can drop any dependence on ξ below to make the proof more readable and concise.

Let

- $p(\mu \to s, t, \pi, x)$ denote the probability given the leader's choise x of reaching state s after t steps starting at μ and following policy π
- $p(\mu \to s, a, t, \pi, x)$ denote the probability under choice x of reaching state s after t steps and then taking action a starting at μ and following policy π
- $p(\mu \to s, a, s', t, \pi, x)$ denote the probability under choice x of reaching state s' after t steps having previously been in state s and having taken action a, starting at μ and following policy π

Assuming $\overline{V}(s)$ is differentiable for all s, we show the following statement by induction.

$$\begin{aligned}
\frac{df(x, \pi_{\lambda,x}^{*}, \xi)}{dx} = &\sum_{s} \mu_x(s) \frac{d \log \mu_x}{dx} \overline{V}(s) + \sum_{t=1}^{n+1} \sum_{s} \sum_{a} \sum_{s}{}' \gamma^t p(\mu_x \to s, a, s', t, \pi_{\lambda,x}^{*}, x) \frac{d \log P_x(s'; s, a)}{dx} \overline{V}(s') \\
&+ \sum_{t=0}^{n} \gamma^t \sum_{s} \sum_{a} p(\mu_x \to s, a, t, \pi_{\lambda,x}^{*}, x) \left(\frac{1}{\lambda} \partial_x A_{\lambda,x}^{\pi_{\lambda,x}^{*}} \overline{Q}(s, a) + \frac{d\overline{r}_x(s, a)}{dx}\right) \\
&+ \gamma^{n+1} \sum_{s} p(\mu_x \to s, t, \pi_{\lambda,x}^{*}, x) \partial_x \overline{V}(s').
\end{aligned} \tag{19}$$

Note that taking $n \to \infty$ then directly proves our claim.

Base case $n = 0$ We prove the statement for $n = 0$

$$\begin{aligned}
\frac{df(x, \pi_x^*, \xi)}{dx} =& \frac{d}{dx} \sum_s \mu(s) \sum_a \pi_x^*(a; s) \overline{Q}(s, a) \\
=& \sum_s \frac{d\mu_x(s)}{dx} \overline{V}(s) + \sum_s \mu_x(s) \sum_a \pi_{\lambda,x}^*(a; s) \left(\frac{1}{\lambda} \partial_x A_{\lambda,x}^{\pi_{\lambda,x}^*} + \partial_x \overline{Q}(s, a) \right) \\
=& \sum_s \frac{d\mu_x(s)}{dx} \overline{V}(s) + \sum_s \mu_x(s) \sum_a \pi_{\lambda,x}^*(a; s) \Bigg(\frac{1}{\lambda} \partial_x A_{\lambda,x}^{\pi_{\lambda,x}^*} + \frac{d\overline{r}_x(s, a)}{dx} \\
& + \gamma \sum_{s'} \left(P_x(s'; s, a) \frac{d \log P_x(s'; s, a)}{dx} \overline{V}(s') + P_x(s'; s, a) \partial_x \overline{V}(s') \right) \Bigg) \\
=& \sum_s \mu_x(s) \frac{d \log \mu_x}{dx} \overline{V}(s) + \sum_{t=1}^{1} \sum_s \sum_a \sum_{s'} \gamma^t p(\mu_x \to s, a, s', t, \pi_{\lambda,x}^*, x) \frac{d \log P_x(s'; s, a)}{dx} \overline{V}(s') \\
& + \sum_{t=0}^{0} \gamma^t \sum_s \sum_a p(\mu_x \to s, a, t, \pi_{\lambda,x}^*, x) \left(\frac{1}{\lambda} \partial_x A_{\lambda,x}^{\pi_{\lambda,x}^*} \overline{Q}(s, a) + \frac{d\overline{r}_x(s, a)}{dx} \right) \\
& + \gamma^1 \sum_s p(\mu_x \to s, t, \pi_{\lambda,x}^*, x) \partial_x \overline{V}(s'),
\end{aligned}$$

where we use the definition of f in the first equality. The second equality follows fromProposition 2 and the product rule. Rearranging terms gives the third equality.

Induction step $n \implies n + 1$ Assuming Equation (19) holds for n, we prove it for $n + 1$.

$$\begin{aligned}
& \sum_s \mu_x(s) \frac{d \log \mu_x}{dx} \overline{V}(s) + \sum_{t=1}^{n+1} \sum_s \sum_a \sum_s{}' \gamma^t p(\mu_x \to s, a, s', t, \pi_{\lambda,x}^*, x) \frac{d \log P_x(s'; s, a)}{dx} \overline{V}(s') \\
& + \sum_{t=0}^{n} \gamma^t \sum_s \sum_a p(\mu_x \to s, a, t, \pi_{\lambda,x}^*, x) \left(\frac{1}{\lambda} \partial_x A_{\lambda,x}^{\pi_{\lambda,x}^*} \overline{Q}(s, a) + \frac{d\overline{r}_x(s, a)}{dx} \right) \\
& + \gamma^{n+1} \sum_s p(\mu_x \to s, t, \pi_{\lambda,x}^*, x) \partial_x \overline{V}(s') \\
=& \sum_s \mu_x(s) \frac{d \log \mu_x}{dx} \overline{V}(s) + \sum_{t=1}^{n+1} \sum_s \sum_a \sum_s{}' \gamma^t p(\mu_x \to s, a, s', t, \pi_{\lambda,x}^*, x) \frac{d \log P_x(s'; s, a)}{dx} \overline{V}(s') \\
& + \sum_{t=0}^{n} \gamma^t \sum_s \sum_a p(\mu_x \to s, a, t, \pi_{\lambda,x}^*, x) \left(\frac{1}{\lambda} \partial_x A_{\lambda,x}^{\pi_{\lambda,x}^*} \overline{Q}(s, a) + \frac{d\overline{r}_x(s, a)}{dx} \right) \\
& + \gamma^{n+1} \sum_s p(\mu_x \to s, t, \pi_{\lambda,x}^*, x) \sum_a \pi_{\lambda,x}^*(a; s) \Bigg(\frac{1}{\lambda} \partial_x A_{\lambda,x}^{\pi_{\lambda,x}^*} + \frac{d\overline{r}_x(s, a)}{dx} \\
& + \gamma \sum_{s'} \left(P_x(s'; s, a) \frac{d \log P_x(s'; s, a)}{dx} \overline{V}(s') + P_x(s'; s, a) \frac{d\overline{V}(s')}{dx} \right) \Bigg) \\
=& \sum_s \mu_x(s) \frac{d \log \mu_x}{dx} \overline{V}(s) + \sum_{t=1}^{n+2} \sum_s \sum_a \sum_s{}' \gamma^t p(\mu_x \to s, a, s', t, \pi_{\lambda,x}^*, x) \frac{d \log P_x(s'; s, a)}{dx} \overline{V}(s') \\
& + \sum_{t=0}^{n+1} \gamma^t \sum_s \sum_a p(\mu_x \to s, a, t, \pi_{\lambda,x}^*, x) \left(\frac{1}{\lambda} \partial_x A_{\lambda,x}^{\pi_{\lambda,x}^*} \overline{Q}(s, a) + \frac{d\overline{r}_x(s, a)}{dx} \right) \\
& + \gamma^{n+2} \sum_s p(\mu_x \to s, t, \pi_{\lambda,x}^*, x) \partial_x \overline{V}(s'),
\end{aligned}$$

which proves our claim. In the first equality we use the definition of $\overline{V}$ and the product rule. The second inequality follows from collecting terms. □

E.7 Auxiliary Results

Proposition 2 (Gradient of Best response Policy)**.** *It holds that*

$$\frac{d\pi^*_{x,\xi}(s,a)}{dx} = \frac{1}{\lambda}\pi^*_{x,\xi}(a;s)\partial_x A^{\pi^*_{x,\xi}}_{\lambda,x,\xi}(s,a).$$

Proof. For a given x, ξ, this result was previoysly shown by [15]. We give a short proof below.

$$\begin{aligned}
\frac{d\pi^*_{x,\xi}(s,a)}{dx} =& \frac{d}{dx}\frac{\exp(Q^*_{\lambda,x,\xi}(s,a)/\lambda)}{\sum'_a \exp(Q^*_{\lambda,x,\xi}(s,a')/\lambda)} \\
=& \frac{\exp(Q^*_{\lambda,x,\xi}(s,a)/\lambda)\cdot\frac{\partial_x Q^*_{\lambda,x,\xi}(s,a)}{\lambda}\sum_{a'}\exp(Q^*_{\lambda,x,\xi}(s,a')/\lambda)}{\left(\sum_{a''}\exp(Q^*_{\lambda,x,\xi}(s,a)/\lambda)\right)^2} \\
&- \frac{\exp(Q^*_{\lambda,x,\xi}(s,a)/\lambda)\sum_{a'}\exp(Q^*_{\lambda,x,\xi}(s,a')/\lambda)\frac{\partial_x Q^*_{\lambda,x,\xi}(s,a')}{\lambda}}{\left(\sum_{a''}\exp(Q^*_{\lambda,x,\xi}(s,a)/\lambda)\right)^2} \\
=& \pi^*(a;s)\frac{\partial_x Q^*_{\lambda,x,\xi}(s,a')}{\lambda} - \frac{\pi^*(a;s)\sum_{a'}\exp(Q^*_{\lambda,x,\xi}(s,a')/\lambda)\frac{\partial_x Q^*_{\lambda,x,\xi}(s,a')}{\lambda}}{\sum_{a''}\exp(Q^*_{\lambda,x,\xi}(s,a)/\lambda)}\cdot \\
=& \frac{1}{\lambda}\pi^*(a;s)\partial_x Q^*_{\lambda,x,\xi}(s,a') - \frac{1}{\lambda}\pi^*(a;s)\sum_{a'}\pi^*(a';s)\partial_x Q^*_{\lambda,x,\xi}(s,a') \\
=& \frac{1}{\lambda}\pi^*(a;s)\left[\partial_x Q^*_{\lambda,x,\xi}(s,a') - \partial_x \mathbb{E}_{a'\sim\pi^*(\cdot;s)}[Q^*_{\lambda,x,\xi}(s,a')]\right] \\
=& \frac{1}{\lambda}\pi^*(a;s)\left(\partial_x Q^*_{\lambda,x,\xi}(s,a) - \partial_x V^*_{\lambda,x,\xi}(s)\right) \\
=& \frac{1}{\lambda}\pi^*(a;s)\partial_x A^*_{\lambda,x,\xi}(s,a).
\end{aligned}$$

The second equality follows from the quotient rule. The third and fourth equality follows from the definition of $\pi^*_{x,\xi}(s,a)$. The remaining equalities leverage the definition of the advantage function.

Note, we can replace the total with the partial derivative of $Q^*(x) = \max_\pi Q(\pi, x)$, due to Rockafellar's thoerem because Q is continuously differentiable in x for all π and Π is compact and convex [9, 56]. □

Proposition 3 (Unbiased advantage derivative estimator)**.** *The output* $\widehat{\partial_x A^{\pi^o_{x,\xi}}_{\lambda,x,\xi}}(s,a)$ *of Algorithm 2 is an unbiased estimate of* $\partial_x A^{\pi^o_{x,\xi}}_{\lambda,x,\xi}(s,a)$*, i.e*

$$\mathbb{E}\left[\widehat{\partial_x A^{\pi^o_{x,\xi}}_{\lambda,x,\xi}}(s,a)\right] = \partial_x A^{\pi^o_{x,\xi}}_{\lambda,x,\xi}(s,a).$$

Proof. We drop any dependence on $x, \xi, \pi^o_{x,\xi}$ for notational clarity. We further emphasize that the trick of truncating a rollout after a geomtrically sampled time to obtain unbiased gradients is commonly used in the RL literature for obtaining unbiased estimates of the standard policy gradient [80].

We show that the estimator $\widehat{\partial_x Q_\lambda}(s,a)$ given by

$$\sum_{t=0}^{T_Q}\frac{d}{dx}r(s_t,a_t) + \frac{\gamma}{1-\gamma}\frac{d}{dx}\log P(s_{T_Q+1};s_{T_Q},a_{T_Q})\sum_{t=T_Q+1}^{T_Q+T'_Q+1}\gamma^{(t-T_Q-1)/2}\left(r(s_t,a_t)+\lambda H(\pi(\cdot;s_t))\right)$$

is unbiased. The same argument then holds for $\widehat{\frac{d}{dx}V_\lambda}(s)$ and implies that $\widehat{\frac{d}{dx}A_\lambda}(s,a)$ is unbiased. First of all, we have:

$$\mathbb{E}\left[\sum_{t=0}^{T'_Q}\gamma^{(t)/2}\left(r(s_t,a_t)+\lambda H(\pi(\cdot;s_t))\right)\right]$$

$$
\begin{aligned}
&= \mathbb{E}_{T_Q'} \mathbb{E}_s^\pi \left[\sum_{t=0}^{T_Q'} \gamma^{(t)/2} \left(r(s_t, a_t) + \lambda H(\pi(\cdot; s_t)) \right) \right] \\
&= \mathbb{E}_s^\pi \left[\mathbb{E}_{T_Q'} \sum_{t=0}^{T_Q'} \gamma^{(t)/2} \left(r(s_t, a_t) + \lambda H(\pi(\cdot; s_t)) \right) \right] \\
&= \mathbb{E}_s^\pi \left[\sum_{t=0}^{\infty} \mathbb{E}_{T_Q'} \left[\mathbb{1}_{t \le T_Q'} \right] \gamma^{(t)/2} \left(r(s_t, a_t) + \lambda H(\pi(\cdot; s_t)) \right) \right] \qquad (20) \\
&= \mathbb{E}_s^\pi \left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + \lambda H(\pi(\cdot; s_t)) \right) \right] \\
&= V_\lambda(s), \qquad (21)
\end{aligned}
$$

where we use Fubini's theorem for eq. (20) and the Dominated Convergence Theorem and the fact that $T_Q' \sim \text{Geo}(1 - \gamma^{0.5})$ in eq. (21). Because T_Q and T_Q' are sampled independently, it immediately follows that

$$
\begin{aligned}
&\mathbb{E}_{T_Q, T_Q'} \mathbb{E}_{s,a}^\pi \Bigg[\sum_{t=0}^{T_Q} \frac{d}{dx} r(s_t, a_t) \\
&\quad + \frac{\gamma}{1-\gamma} \frac{d}{dx} \log P(s_{T_Q+1}; s_{T_Q}, a_{T_Q}) \sum_{t=T_Q+1}^{T_Q + T_Q' + 1} \gamma^{(t - T_Q - 1)/2} \left(r(s_t, a_t) + \lambda H(\pi(\cdot; s_t)) \right) \Bigg] \\
&= \mathbb{E}_{T_Q} \mathbb{E}_{s,a}^\pi \Bigg[\underbrace{\sum_{t=0}^{T_Q} \frac{d}{dx} r(s_t, a_t)}_{(1)} + \underbrace{\frac{\gamma}{1-\gamma} \frac{d \log P(s_{T_Q+1}; s_{T_Q}, a_{T_Q})}{dx} V_\lambda^\pi(s_{T_Q})}_{(2)} \Bigg].
\end{aligned}
$$

We seperately show that the two summands are unbaised estimates. Then by linearity of expectation the result follows. For (1) using Fubini's theorem and Dominated Convergence Theorem it holds that

$$
\begin{aligned}
\mathbb{E}_{T_Q} \mathbb{E}_{s,a}^\pi \left[\sum_{t=0}^{T_Q} \frac{d}{dx} r(s_t, a_t) \right] &= \mathbb{E}_{s,a}^\pi \left[\sum_{k=0}^{\infty} (1-\gamma) \gamma^k \sum_{t=0}^{k} \frac{d}{dx} r(s_t, a_t) \right] \\
&= (1-\gamma) \mathbb{E}_{s,a}^\pi \left[\sum_{t=0}^{\infty} \sum_{k=t}^{\infty} \gamma^k \frac{d}{dx} r(s_t, a_t) \right] \\
&= (1-\gamma) \mathbb{E}_{s,a}^\pi \left[\sum_{t=0}^{\infty} \frac{\gamma^t}{1-\gamma} \frac{d}{dx} r(s_t, a_t) \right] \\
&= \mathbb{E}_{s,a}^\pi \left[\sum_{t=0}^{\infty} \gamma^t \frac{d}{dx} r(s_t, a_t) \right].
\end{aligned}
$$

Similarly for (2), we have using Fubini and Dominated Convergence Theorem that

$$
\begin{aligned}
&\mathbb{E}_{T_Q} \mathbb{E}_{s,a}^\pi \left[\frac{\gamma}{1-\gamma} \frac{d \log P(s_{T_Q+1}; s_{T_Q}, a_{T_Q})}{dx} V_\lambda^\pi(s_{T_Q}) \right] \\
&= \frac{\gamma}{1-\gamma} \mathbb{E}_{s,a}^\pi \mathbb{E}_{T_Q} \left[\sum_{t=0}^{\infty} \mathbb{1}_{t = T_Q} \frac{d \log P(s_{t+1}; s_t, a_t)}{dx} V_\lambda^\pi(s_t) \right] \\
&= \mathbb{E}_{s,a}^\pi \left[\sum_{t=0}^{\infty} \gamma^{t+1} \frac{d \log P(s_{t+1}; s_t, a_t)}{dx} V_\lambda^\pi(s_t) \right].
\end{aligned}
$$

Plugging these results back in, we have

$$\begin{aligned}
&\mathbb{E}_{T_Q}\mathbb{E}^{\pi}_{s,a}\left[\sum_{t=0}^{T_Q}\frac{d}{dx}r(s_t,a_t)+\frac{\gamma}{1-\gamma}\frac{d\log P(s_{T_Q+1};s_{T_Q},a_{T_Q})}{dx}V^{\pi}_{\lambda}(s_{T_Q})\right]\\
=&\mathbb{E}^{\pi}_{s,a}\left[\sum_{t=0}^{\infty}\gamma^t\frac{d}{dx}r(s_t,a_t)+\gamma^{t+1}\frac{d\log P(s_{t+1};s_t,a_t)}{dx}V^{\pi}_{\lambda}(s_t)\right]\\
=&\partial_x Q^{\pi}_{\lambda}(s,a),
\end{aligned}$$

which proves the proposition. □

Proposition 4 (Unbiased gradient estimator for F)**.** *The gradient estimator described in Algorithm 5 is unbiased for the given policy $\pi^o_{x,\xi}$.*

Proof. We need to show that:

$$\begin{aligned}
&\mathbb{E}\Bigg[\underbrace{\left(\sum_{t=0}^{T}\frac{d}{dx}\overline{r}(s_t,a_t)\right)}_{(1)}+\underbrace{\frac{1}{\lambda(1-\gamma)}\partial_x\widehat{A^{\pi^o_{x,\xi}}_{\lambda,x,\xi}}(s_T,a_T)\sum_{t'=T}^{T+T'}\gamma^{(t-T)/2}\overline{r}(s_{t'},a_{t'})}_{(2)}\\
&+\underbrace{\frac{1}{1-\gamma}\partial_x\log P(s_T,a_{T-1},s_{T-1})\sum_{t'=T}^{T+T'}\gamma^{(t'-T)/2}\overline{r}(s_{t'},a_{t'})}_{(3)}\Bigg]\\
=&\mathbb{E}_{\xi}\left[\mathbb{E}^{\pi^o_{x,\xi}}_{s_0\sim\mu}\left[\sum_{t=0}^{\infty}\gamma^t\left(\frac{1}{\lambda}\partial_x A^{\pi^o_{x,\xi}}_{\lambda,x,\xi}(s_t,a_t)\overline{Q}(s_t,a_t)+\frac{d}{dx}\overline{r}(s_t,a_t)+\partial_x\log P_{x,\xi}(s_t,a_{t-1},s_{t-1})\overline{V}(s_t)\right)\right]\right].
\end{aligned}$$

We can show the claim seperately for (1), (2) and (3). Note for (1) and (3) the claim directly follows from the proof of Proposition 3. And the proof for (2) works almost identical to the one for (3), relying on the fact that a truncation via a geometric distribution is identical to an infinite trajectory with a discount factor. □

E.8 Convergence Results for Popular RL Algorithms

For the next Proposition, consider the following soft Bellmann optimality operator, which has been shown to be a contraction [20, 51].

$$\left(\mathcal{T}^*_{\lambda}V_{\lambda}\right)(s):=\lambda\log\left(\sum_{a\in\mathcal{A}}\exp\left(\frac{r(s,a)+\gamma\mathbb{E}_{s'|s,a}\left[V_{\lambda}(s')\right]}{\lambda}\right)\right). \tag{22}$$

Using Equation (22), one can define a standard soft value iteration algorithm (see Algorithm 3 in Appendix D). We show soft value iteration satisfies Assumption 3.2.

Proposition 5. *Algorithm 3 converges, such that $\left\|\pi^*_{x,\xi}-\pi^o_{x,\xi}\right\|^2_{\infty}\leq\delta^2$ after T iterations, where $T=\mathcal{O}(\log 1/\delta)$.*

Proof. From [49][Lemma 24] we have

$$\left\|\pi^o-\pi^*\right\|_{\infty}\leq\left\|\pi^o-\pi^*\right\|_1\leq\frac{1}{\lambda}\left\|Q^T_{\lambda}-Q^*_{\lambda}\right\|_{\infty}.$$

Moreover

$$\frac{1}{\lambda}\left\|Q^T_{\lambda}-Q^*_{\lambda}\right\|_{\infty}\leq\frac{1}{\lambda}\left\|V^T_{\lambda}-V^*_{\lambda}\right\|_{\infty}\leq\frac{\gamma^T}{\lambda}\left\|V^*_{\lambda}\right\|\leq\frac{\gamma^T}{\lambda(1-\gamma)}\left(\overline{R}+\lambda\log|\mathcal{A}|\right),$$

where we use the contraction property shown in [20, 51] and the fact that we instantiate V_{λ} with 0, such that no value iterate can ever be larger than higher than $\left(\overline{R}+\lambda\log|\mathcal{A}|\right)$. The claim follows from $\delta\leq\mathcal{O}(\gamma^T)$. □

As soft value iteration assumes knowledge of the transition function and scales badly when the state and action space are large, in practice stochastic methods such as soft Q-learning are used instead. For this method, consider the soft Bellman state-action optimality operator [3, 30]:

$$\left(\mathcal{T}_\lambda^* Q_\lambda\right)(s,a) := r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\lambda \log \left(\sum_{a' \in \mathcal{A}} \exp\left(\frac{Q_\lambda(s,a')}{\lambda} \right) \right) \right]. \tag{23}$$

We can use Equation (23) to run soft Q-learning, as described in Algorithm 4 in Appendix D. Equivalently to soft value iteration, we can show soft Q-learning satisfies Assumption 3.2.

Proposition 6. *Let π_B be sufficiently exploratory, such that the induced Markov chain is ergodic. Then soft Q-learning converges, such that $\mathbb{E}_o \left[\left\| \pi^*_{x,\xi} - \pi^o_{x,\xi} \right\|_\infty^2 \right] \leq \delta^2$ after T iterations, where $T = \mathcal{O}(\frac{\log(1/\delta)}{\delta^2})$.*

We use the following Theorem from [54] to prove our claim:

Theorem [54] Let $x \in \mathbb{R}^d$, and $F : \mathbb{R}^d \to \mathbb{R}^d$ be an operator. We use F_i to denote the i 'th entry of F. We consider the following stochastic approximation scheme that keeps updating $x(t) \in \mathbb{R}^d$ starting from $x(0)$ being the all zero vector,

$$\begin{aligned} x_i(t+1) &= x_i(t) + \alpha_t \left(F_i(x(t)) - x_i(t) + w(t) \right) && \text{for } i = i_t, \\ x_i(t+1) &= x_i(t) && \text{for } i \neq i_t, \end{aligned}$$

where $i_t \in \{1, \ldots, d\}$ is a stochastic process adapted to a filtration $\mathcal{F}_t$, and $w(t)$ is some noise. Assume the following:

Assumption 1 (Contraction) (a) Operator F is γ contraction in $\|\cdot\|_\infty$, i.e. for any $x, y \in \mathbb{R}^d$, $\|F(x) - F(y)\|_\infty \leq \gamma \|x - y\|_\infty$. (b) There exists some constant $C > 0$ s.t. $\|F(x)\|_\infty \leq \gamma \|x\|_\infty + C, \forall x \in \mathbb{R}^d$.

Assumption 2 (Martingale Difference Sequence) $w(t)$ is $\mathcal{F}_{t+1}$ measurable and satisfies $\mathbb{E} w(t) \mid \mathcal{F}_t = 0$. Further, $|w(t)| \leq \bar{w}$ almost surely for some constant $\bar{w}$.

Assumption 3 (Sufficient Exploration) There exists a $\sigma \in (0,1)$ and positive integer, τ, such that, for any $i \in \mathcal{N}$ and $t \geq \tau, \mathbb{P}\left(i_t = i \mid \mathcal{F}_{t-\tau}\right) \geq \sigma$.

Suppose Assumptions 1,2 and 3 hold. Further, assume there exists constant $\bar{x} \geq \|x^*\|_\infty$ s.t. $\forall t, \|x(t)\|_\infty \leq \bar{x}$ almost surely. Let the step size be $\alpha_t = \frac{h}{t+t_0}$ with $t_0 \geq \max(4h, \tau)$, and $h \geq \frac{2}{\sigma(1-\gamma)}$. Then, with probability at least $1 - \delta$,

$$\|x(T) - x^*\|_\infty \leq \frac{12\bar{\epsilon}}{1-\gamma} \sqrt{\frac{(\tau+1)h}{\sigma}} \sqrt{\frac{\log\left(\frac{2(\tau+1)T^2 n}{\delta}\right)}{T+t_0}} + \frac{4}{1-\gamma} \max\left(\frac{16\bar{\epsilon} h \tau}{\sigma}, 2\bar{x}\left(\tau + t_0\right) \right) \frac{1}{T+t_0},$$

where $\bar{\epsilon} = 2\bar{x} + C + \bar{w}$.

Proof. Our algorithm can be seen as a stochastic approximation scheme where we update Q asynchronously just like x above in the following way

$$\begin{aligned} Q_{s_t,a_t}(t+1) &= Q_{s_t,a_t}(t) + \alpha_t \left(F_{s_t,a_t}(Q(t)) - Q_{s_t,a_t}(t) + w_t \right) \\ Q_{s,a}(t+1) &= Q_{s,a}(t) && \text{for } s, a \neq s_t, a_t, \end{aligned}$$

where

$$F_{s_t,a_t}(Q) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\lambda \log \left(\sum_{a' \in \mathcal{A}} \exp\left(\frac{Q(s,a')}{\lambda} \right) \right) \right],$$

and the errors:

$$\begin{aligned} w_t =& r(s_t,a_t) + \gamma \lambda \log \left(\sum_{a' \in \mathcal{A}} \exp\left(\frac{Q_\lambda(s_{t+1},a')}{\lambda} \right) \right) \\ & - r(s_t,a_t) + \gamma \mathbb{E}_{s' \sim P(\cdot;s_t,a_t)} \left[\lambda \log \left(\sum_{a' \in \mathcal{A}} \exp\left(\frac{Q_\lambda(s_{t+1},a')}{\lambda} \right) \right) \right]. \end{aligned}$$

We now show that F satisfies the assumptions of the Theorem from [54] and use the result to prove our own claim.

In the following we let $\mathcal{F}_t$ be the $\sigma-$algebra generated by the random variables $(s_0, a_0, \cdots, s_t, a_t)$.

First we restate the following identity from [51]

$$\begin{aligned} T^*_\lambda(Q)(s,a) &= r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\lambda \log \left(\sum_{a' \in \mathcal{A}} \exp\left(\frac{Q(s,a')}{\lambda} \right) \right) \right] \\ &= r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\max_\pi \langle Q(\cdot, s'), \pi \rangle + \lambda H(\pi; s') \right]. \end{aligned}$$

We use it to show that T^*_λ is a contraction. Indeed we have:

$$\begin{aligned} & \left\| T^*_\lambda(Q_1) - T^*_\lambda(Q_2) \right\|_\infty \\ =& \Bigg\| r(s,a) + \gamma \max_\pi \sum_{s'} P(s';s,a) \left(\langle Q_1(\cdot, s'), \pi \rangle + \lambda H(\pi; s') \right) \\ & - r(s,a) - \gamma \max_\pi \sum_{s'} P(s';s,a) \left(\langle Q_2(\cdot, s'), \pi \rangle + \lambda H(\pi; s') \right) \Bigg\|_\infty \\ \leq& \gamma \left\| \max_\pi \sum_{s'} P(s';s,a) \left(\langle Q_1(\cdot, s'), \pi \rangle - \langle Q_2(\cdot, s'), \pi \rangle \right) \right\|_\infty \\ \leq& \gamma \left\| Q_1 - Q_2 \right\|_\infty. \end{aligned}$$

Moreover, it holds that

$$F(Q) \leq \overline{R} + \gamma \left\| Q \right\|_\infty + \lambda \log |\mathcal{A}|.$$

So we can set $C = \overline{R} + \lambda \log |\mathcal{A}|$

Next we note that w_t is $F_{t+1}-$measurable (it depends on s_{t+1}) and that

$$\mathbb{E}[w_t | \mathcal{F}_t] = 0.$$

Moreover $w(t)$ is bounded by $\overline{w} = \frac{2\gamma(\overline{R} + \lambda \log |\mathcal{A}|)}{1-\gamma}$.

Further we have assumed that the behavioural policy π_B is sufficiently exploratory. Let $\tilde{\mu}$ be the corresponding stationary distribution, $\mu_{\min} = \inf_{s,a} \tilde{\mu}(s,a)$ and t_{mix} the mixing time. Then [54] show that for $\sigma = \frac{1}{2}\mu_{\min}$ and $\tau = \lceil \log_2(\frac{2}{\mu_{\min}}) \rceil t_{mix}$ it holds that

$$\forall s \in \mathcal{S}, a \in \mathcal{A}, \forall t \geq \tau : \mathbb{P}(s_t, a_t = s, a | \mathcal{F}_{t-\tau}) \geq \sigma. \tag{24}$$

Moreover, we note that $Q(t)$ and Q^*_λ are bound by $\overline{x} = \frac{\overline{R} + \lambda \log |\mathcal{A}|}{1-\gamma}$.

Using the Theorem from [54] we thus have the following result:

Let $\alpha_t = \frac{h}{t+t_0}$ with $t_0 \geq \max\left(4h, \left\lceil \log_2 \frac{2}{\mu_{\min}} \right\rceil t_{\text{mix}}\right)$ and $h \geq \frac{4}{\mu_{\min}(1-\gamma)}$. Then, with probability at least $1-p$,

$$\begin{aligned} & \left\| Q(T) - Q^*_\lambda \right\|_\infty \leq \\ & \leq \frac{60(\bar{R} + \lambda \log |\mathcal{A}|)}{(1-\gamma)^2} \sqrt{\frac{2\left(\left\lceil \log_2 \frac{2}{\mu_{\min}} \right\rceil t_{\text{mix}} + 1 \right) h}{\mu_{\min}}} \sqrt{\frac{\log\left(\frac{2\left(\left\lceil \log_2 \frac{2}{\mu_{\min}} \right\rceil t_{\text{mix}} + 1 \right) T^2 |\mathcal{S}\|\mathcal{A}|}{p} \right)}{T + t_0}} \\ & + \frac{4(\bar{R} + \lambda \log |\mathcal{A}|)}{(1-\gamma)^2} \max\left(\frac{160h \left\lceil \log_2 \frac{2}{\mu_{\min}} \right\rceil t_{\text{mix}}}{\mu_{\min}}, 2\left(\left\lceil \log_2 \frac{2}{\mu_{\min}} \right\rceil t_{\text{mix}} + t_0 \right) \right) \frac{1}{T + t_0}. \end{aligned}$$

We denote the bound above by **(A)**. Let us choose $p = \delta^2$.[6] With probability p, $\|Q(T) - Q^*_\lambda\|_\infty$ is not bounded by **(A)**. However, it is always upper bounded by $\frac{2(\bar{R}+\lambda \log(|\mathcal{A}|))}{1-\gamma}$.

Setting $T = \mathcal{O}(\frac{\log(1/\delta)}{\delta^2})$, we get

$$
\begin{aligned}
\textbf{(A)} =& \mathcal{O}\left(\sqrt{\frac{\log\left(\frac{T^2}{\delta^2}\right)}{T}} + \frac{1}{T}\right) \\
=& \mathcal{O}\left(\sqrt{\frac{\log\left(\frac{T}{\delta}\right)}{T}}\right) \\
\underset{T=\frac{\log(1/\delta)}{\delta^2}}{=}& \mathcal{O}\left(\sqrt{\frac{\log\left(\frac{\frac{\log(1/\delta)}{\delta^2}}{\delta}\right)}{\frac{\log(1/\delta)}{\delta^2}}}\right) \\
=& \mathcal{O}\left(\delta\sqrt{\frac{\log\left(\frac{\log(1/\delta)}{\delta^3}\right)}{\log\left(1/\delta\right)}}\right) \\
=& \mathcal{O}\left(\delta\sqrt{\frac{-3\log\left(\delta\right) + \log\left(\log\left(1/\delta\right)\right)}{\log\left(1/\delta\right)}}\right) \\
=& \mathcal{O}\left(\delta\sqrt{\frac{-3\log\left(\delta\right)}{-\log\left(\delta\right)} + \frac{\log\left(\log\left(1/\delta\right)\right)}{\log\left(1/\delta\right)}}\right).
\end{aligned}
$$

Since $\delta < 1$, $\log(1/\delta) > 0$ and for $x > 0$ it holds that $\frac{\log(x)}{x} < 1/e$, such that we get:

$$
\begin{aligned}
\textbf{(A)} \leq& \mathcal{O}\left(\delta\sqrt{3 + \frac{1}{e}}\right) \\
=& \mathcal{O}\left(\delta\right).
\end{aligned}
$$

Using [49][Lemma 24], we arrive at:

$$
\begin{aligned}
&\mathbb{E}_o\left[\|\pi^o - \pi^*\|^2_\infty\right] \\
\leq& 4\mathbb{E}_o\left[\left\|Q^T_\lambda - Q^*_\lambda\right\|^2_\infty\right] \\
\leq& 4\left((1-p)\textbf{(A)}^2 + p\left(\frac{2(\bar{R}+\lambda\log(|\mathcal{A}|))}{1-\gamma}\right)^2\right) \\
=& \mathcal{O}(\delta^2).
\end{aligned}
$$

□

A popular class of RL algorithms are policy gradient methods such as REINFORCE [68]. For the entropy-regularised problem, it generally makes sense to choose a softmax parametrization for the policy, as we know $\pi^*_{x,\xi}(s;a) \propto \exp(Q^*_{\lambda,x,\xi}(s,a)/\lambda)$ [49]. We defer the details to Algorithm 6 in Appendix D and present the following convergence result, which shows using vanilla policy gradient for the lower level also fulfills Assumption 3.2—at least asymptotically.

Proposition 7. *Vanilla policy gradient with softmax parameterization converges, such that* $\forall\delta, \exists T, \forall t \geq T : \left\|\pi^*_{x,\xi} - \pi^o_{t,x,\xi}\right\| \leq \delta^2$, *where* $\pi^o_{t,x,\xi}$ *is the computed policy after* t *iterations.*

[6] Note that $\delta < 1$, as it represents the distance between two softmax policies under the supremum norm.

Proof. As in most proofs we drop the subscripts for x, ξ. The proof is an adaptation of the one presented in [49][Lemma 16]. We denote by π_t the iterates of the policies of the algorithm and by $V_\lambda^{\pi_t}(\mu)$ the corresponding value function with starting distribution μ. It can be shown that V_λ^π is s-smooth for some s [49]. Choosing a stepsize of $1/s$, we have that the value functions increase monotonically, i.e.

$$\forall t : V_\lambda^{\pi_{t+1}}(\mu) \geq V_\lambda^{\pi_t}(\mu).$$

At the same time, it holds that:

$$V_\lambda^{\pi_t}(\mu) \leq \frac{\overline{R} + \lambda \log \mathcal{A}}{1-\gamma}.$$

By monotone convergence it thus follows that $V_\lambda^{\pi_t}(\mu) \to V_\lambda^*(\mu)$, where $V_\lambda^*(\mu)$ is the maximum possible value.

Since $\pi_t \in \Delta(\mathcal{A})^{|\mathcal{S}|}$ and $\Delta(\mathcal{A})^{|\mathcal{S}|}$ is compact it follows that $\{\pi_t\}_t$ has a convergent subsequence $\{\pi_{t_k}\}_k$. Denote by π^* the limit of this subsequence. It has to hold that $V_\lambda^{\pi^*}(\mu) = V_\lambda^*(\mu)$ and thus π^* is the optimal policy.

Now assume that $\{\pi_t\}_t$ does not converge to π^*. In that case

$$\exists \epsilon, \forall t, \exists t' \geq t : \|\pi^* - \pi_{t'}\|_\infty > \epsilon.$$

Note that due to entropy regularization $V_\lambda^*(\mu)$ is the unique maximum. This means that

$$\exists \kappa : \max\{V_\lambda^\pi \,|\, \|\pi - \pi^*\|_\infty \geq \epsilon\} + \kappa < V_\lambda^*.$$

It follows then that

$$\forall t, \exists t' \geq t : \left\| V_\lambda^{\pi^*} - V_\lambda^{\pi_{t'}} \right\|_\infty > \kappa,$$

which implies $V_\lambda^{\pi_t}(\mu)$ does not converge to V^*—a contradiction to our conclusion above. It therefore has to hold that $\pi_t \to \pi^*$. □

The asymptotic guarantee of Vanilla Policy Gradient can be improved to non-asymptotic by using Natural Policy Gradient, as introduced by [42]. We restate the following result from [12].

Proposition 8 (Linear convergence of exact entropy-regularized NPG, [12])**.** *For any learning rate $0 < \eta \leq (1-\gamma)/\tau$, the entropy-regularized NPG updates (18) satisfy*

$$\left\| Q_\lambda^\star - Q_\lambda^{(t+1)} \right\|_\infty \leq C_1 \gamma (1-\eta\lambda)^t$$
$$\left\| \log \pi_\lambda^\star - \log \pi^{(t+1)} \right\|_\infty \leq 2C_1 \lambda^{-1} (1-\eta\lambda)^t,$$

for all $t \geq 0$, where

$$C_1 := \left\| Q_\lambda^\star - Q_\lambda^{(0)} \right\|_\infty + 2\lambda \left(1 - \frac{\eta\lambda}{1-\gamma}\right) \left\| \log \pi_\lambda^\star - \log \pi^{(0)} \right\|_\infty.$$

F Implementation Details

F.1 Baseline Algorithms

F.1.1 Adaptive Model Design [15]

As noted in Section 5, the Adaptive Model Design (AMD) algorithm [15] was proposed for the Regularized Markov Design (RMD) problem which is a special case of Contextual Bilevel Reinforcement Learning. In particular, when $\mathbb{P}_\xi$ is a Dirichlet distribution CB-RL reduces to the RMD problem. To account for this difference, we modify the AMD algorithm (Algorithm 2 in [15]) as described in Algorithm 9. We denote the upper-level reward and value functions with the superscript u in the algorithm.

F.1.2 Zero-Order Algorithm

Algorithm 10 defines the zero-order gradient estimation algorithm described in Section 5. We parametrize the perturbation constant to decrease with the number of iterations such as $u_t = \frac{C}{t}$ where C is a positive constant.

Algorithm 9 (Modified) Adaptive Model Design

Input: Iterations T, Inner iterations: K, Learning rate α, Regularization λ, gradient of the pre-learned model $\nabla_x \log P$, gradient of the reward function $\nabla_x r$
Initialize x_0, Q_0, $\nabla_x Q_0$, and $\tilde{Q}_0$
for $t = 0$ to $T-1$ **do**
  $\xi \sim \mathbb{P}_\xi$
  for $k = 0$ to $K-1$ **do**
    $\pi_{x_t,\xi} \leftarrow \exp(\lambda Q_k(s,\cdot))$
    Calculate $V_k, \nabla_{x_t} V_k, V_k^U, \nabla_{x_t} A_k, A_k^u, \tilde{V}_k$
    $Q_{k+1} \leftarrow \mathcal{T}_{r,\gamma}(V_k)$
    $\nabla_{x_t} Q_{k+1} = \mathcal{T}_{\nabla_{x_t} r,\gamma}(\nabla_{x_t} V_k + V_k \nabla_{x_t} \log P)$
    $Q^u_{k+1} = \mathcal{T}_{r_u,\gamma_u}(V_k^u)$
    $\tilde{Q}_{k+1} \leftarrow \mathcal{T}_{\nabla_{x_t} r^u + \lambda A_k^u \nabla_{x_t} A_k}(\tilde{V}_k + V_k^u \nabla_{x_t} \log P)$
  end for
  Set $\widehat{\frac{dF}{dx}} = \tilde{V}_K$
  $x_{t+1} \leftarrow x_t + \alpha \widehat{\frac{dF}{dx}}$
  Reinitialize $Q_0 \leftarrow Q_K$, $\nabla_x Q_0 \leftarrow \nabla_x Q_K$, and $\tilde{Q}_0 \leftarrow \tilde{Q}_K$
end for
Output: Optimised parameter x_T

Algorithm 10 Zero-Order Algorithm

Input: Iterations T, Learning rate α, Regularization λ
Initialize x_0
for $t = 0$ to $T-1$ **do**
  $\xi \sim \mathbb{P}_\xi$
  Sample $z \sim N(0, I_{d_x})$
  $\pi^o_{x_t,\xi} \leftarrow \texttt{OraclePolicy}(x_t, \xi)$
  $\pi^o_{x_t+u_t*z,\xi} \leftarrow \texttt{OraclePolicy}(x_t + z u_t, \xi)$
  Set $\widehat{\frac{dF}{dx}} = \frac{f(x+u_t*z, \pi^*_{x+u_t*z,\xi}, \xi) - f(x, \pi^*_{x,\xi}, \xi)}{u_t} z$
  $x_{t+1} \leftarrow x_t + \alpha \widehat{\frac{dF}{dx}}$
end for
Output: $\hat{x}_T \sim U(\{x_0, \ldots, x_{T-1}\})$

F.2 Four Rooms

F.2.1 Implementation Details

We parametrize the penalty function $\tilde{r}$ as the softmax transformation of $x \in \mathbb{R}^{d_s+1}$ where the i-th entry of x corresponds to the i-th cell in the state space $\mathcal{S}$ and the additional dimension $d_s + 1$ is used to allocate the penalties not effective and also excluded from the penalty term received by the leader at the end of each episode. In particular,

$$\tilde{r}(s,a) = -0.2 * \mathrm{softmax}(s; x),$$

where $\mathrm{softmax}(s; x)$ denotes the value of the softmax transformation of x at the entry corresponding to the state s. Note that this parametrization explicitly restricts the maximum available budget for penalties to -0.2.

F.2.2 Hyperparameters

For the upper-level optimization problem, we use gradient norm clipping of 1.0. The learning rate for each algorithm has been chosen as the best performing one from $[1.0, 0.5, 0.1, 0.05, 0.01]$ individually. Additionally, we tune the parameter C for the Zero-order algorithm on the values $[0.1, 0.5, 1.0, 2.0, 5.0]$. For Hyper Policy Gradient Descent, we sample $10,000$ environment steps for each gradient calculation.

F.2.3 Additional Figures

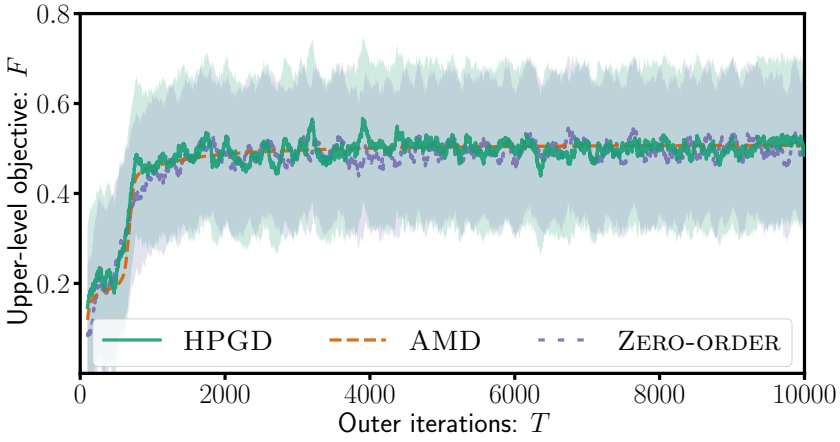


Figure 4: Upper-level objective values, F, over the number of outer iterations for hyperparameters $\lambda = 0.001$ and $\beta = 3.0$

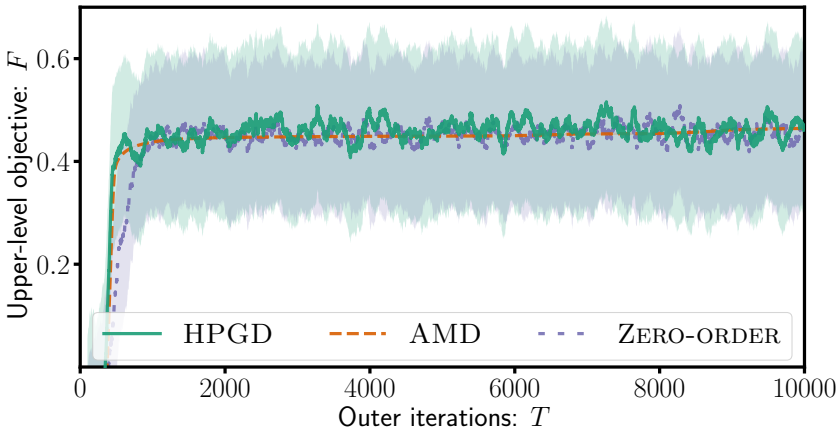


Figure 5: Upper-level objective values, F, over the number of outer iterations for hyperparameters $\lambda = 0.001$ and $\beta = 5.0$

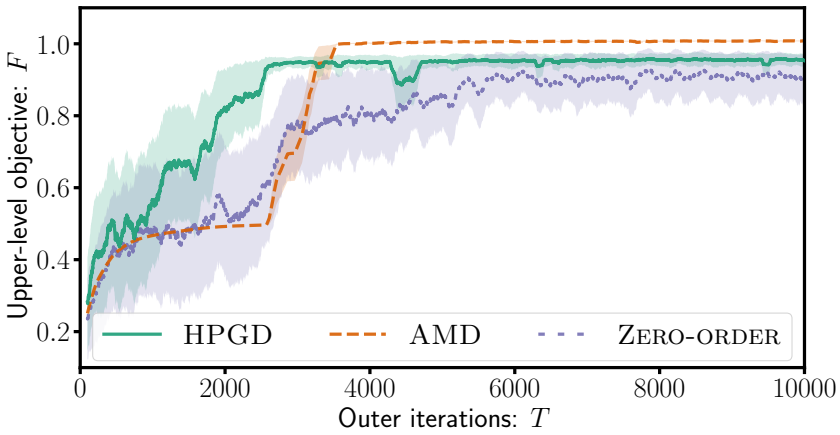


Figure 6: Upper-level objective values, F, over the number of outer iterations for hyperparameters $\lambda = 0.003$ and $\beta = 1.0$

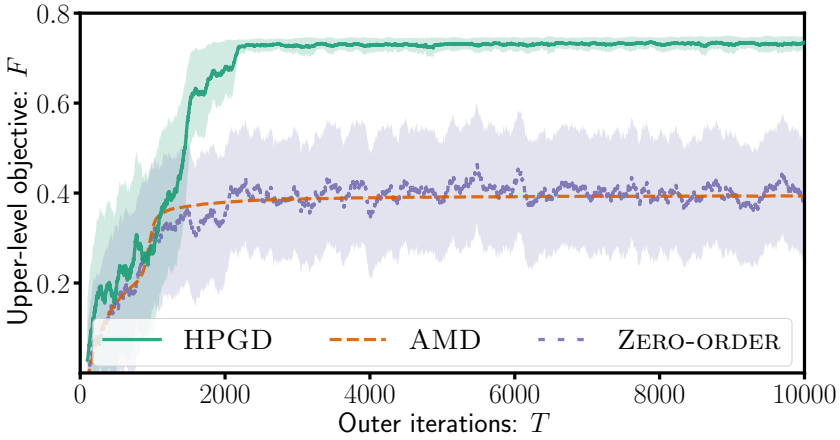


Figure 7: Upper-level objective values, F, over the number of outer iterations for hyperparameters $\lambda = 0.003$ and $\beta = 3.0$

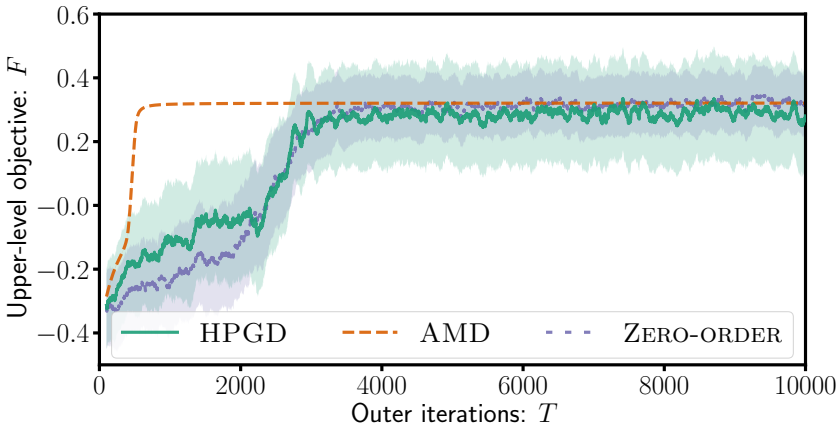


Figure 8: Upper-level objective values, F, over the number of outer iterations for hyperparameters $\lambda = 0.003$ and $\beta = 5.0$

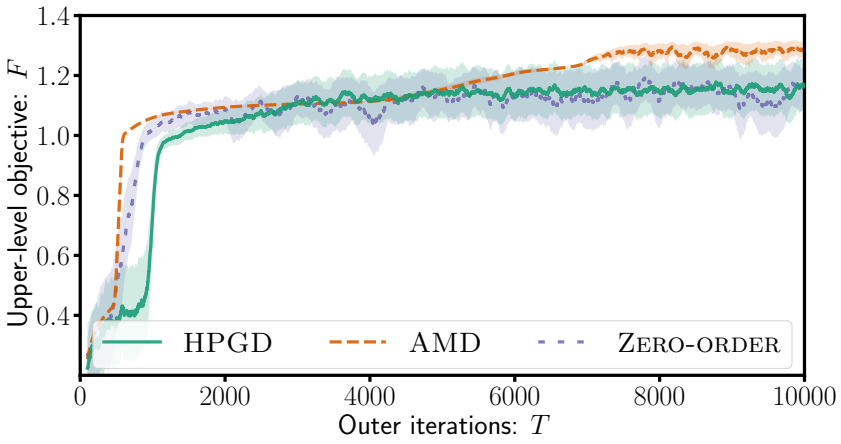


Figure 9: Upper-level objective values, F, over the number of outer iterations for hyperparameters $\lambda = 0.005$ and $\beta = 1.0$

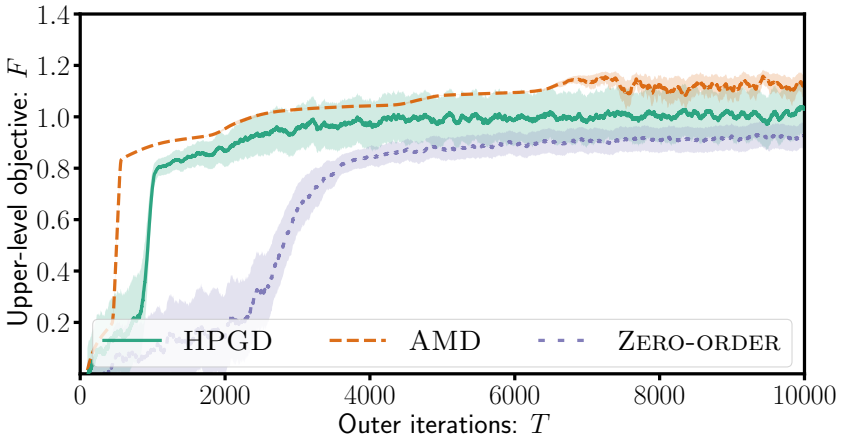


Figure 10: Upper-level objective values, F, over the number of outer iterations for hyperparameters $\lambda = 0.005$ and $\beta = 3.0$

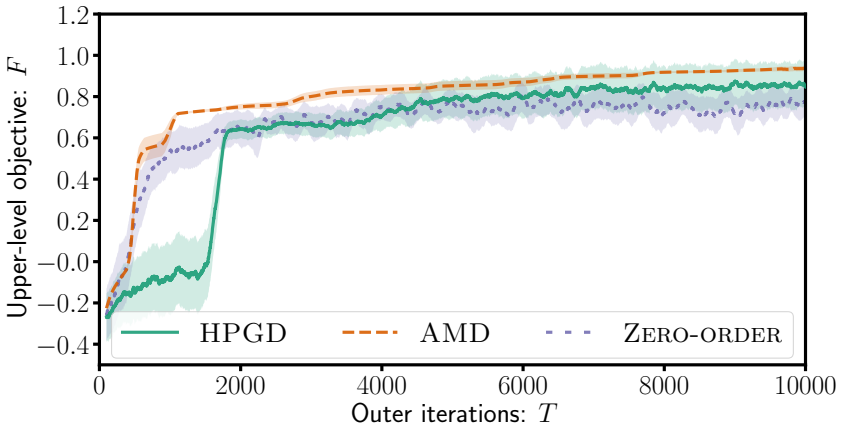


Figure 11: Upper-level objective values, F, over the number of outer iterations for hyperparameters $\lambda = 0.005$ and $\beta = 5.0$

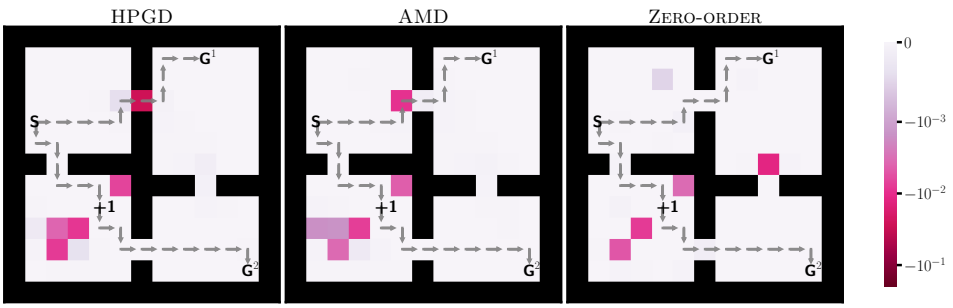


Figure 12: Reward penalties given to the lower-level agent in each state of the Four-Rooms problem optimized by the HPGD, AMD, and Zero-Order, respectively, for hyperparameters $\lambda = 0.001$ and $\beta = 3.0$

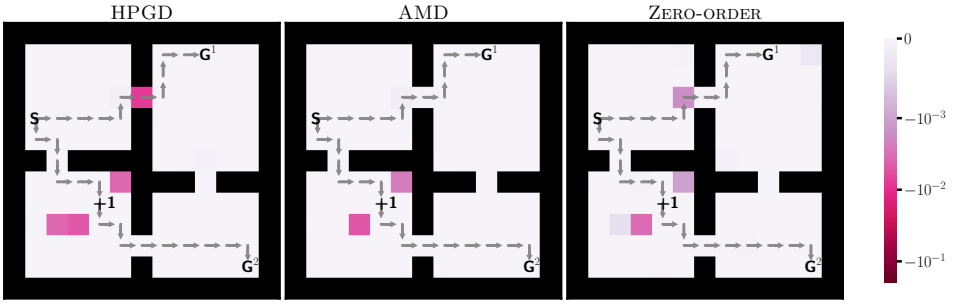


Figure 13: Reward penalties given to the lower-level agent in each state of the Four-Rooms problem optimized by the HPGD, AMD, and Zero-Order, respectively, for hyperparameters $\lambda = 0.001$ and $\beta = 5.0$

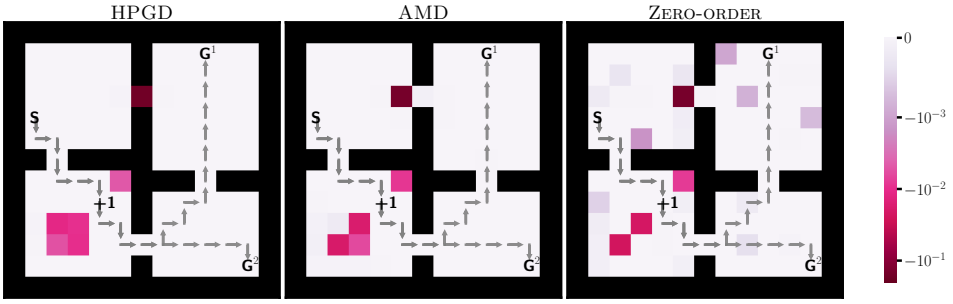


Figure 14: Reward penalties given to the lower-level agent in each state of the Four-Rooms problem optimized by the HPGD, AMD, and Zero-Order, respectively, for hyperparameters $\lambda = 0.003$ and $\beta = 1.0$

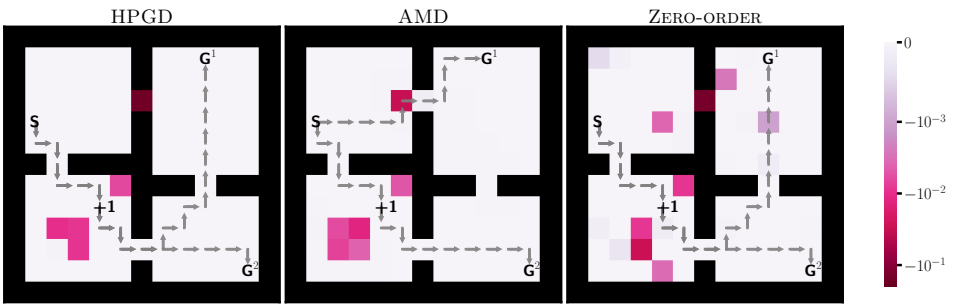


Figure 15: Reward penalties given to the lower-level agent in each state of the Four-Rooms problem optimized by the HPGD, AMD, and Zero-Order, respectively, for hyperparameters $\lambda = 0.003$ and $\beta = 3.0$

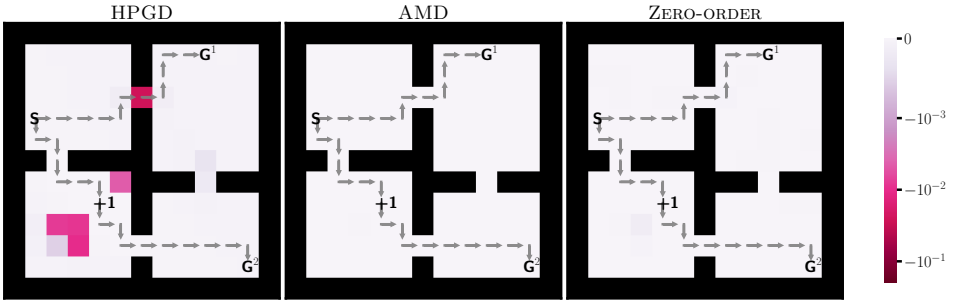


Figure 16: Reward penalties given to the lower-level agent in each state of the Four-Rooms problem optimized by the HPGD, AMD, and Zero-Order, respectively, for hyperparameters $\lambda = 0.003$ and $\beta = 5.0$

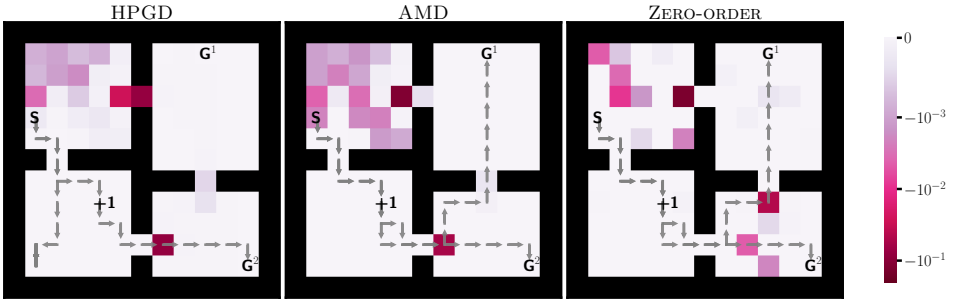


Figure 17: Reward penalties given to the lower-level agent in each state of the Four-Rooms problem optimized by the HPGD, AMD, and Zero-Order, respectively, for hyperparameters $\lambda = 0.005$ and $\beta = 1.0$

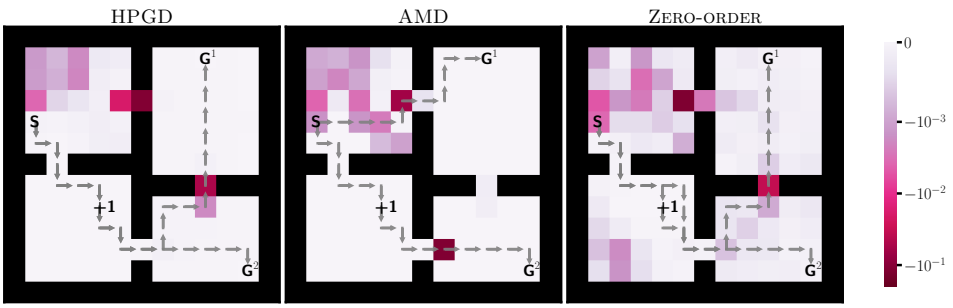


Figure 18: Reward penalties given to the lower-level agent in each state of the Four-Rooms problem optimized by the HPGD, AMD, and Zero-Order, respectively, for hyperparameters $\lambda = 0.005$ and $\beta = 3.0$

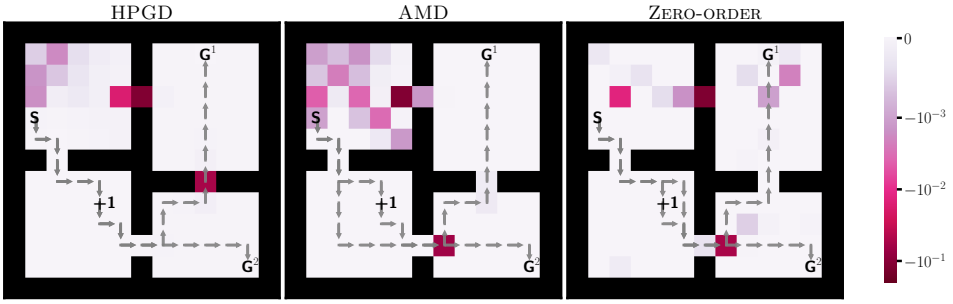


Figure 19: Reward penalties given to the lower-level agent in each state of the Four-Rooms problem optimized by the HPGD, AMD, and Zero-Order, respectively, for hyperparameters $\lambda = 0.005$ and $\beta = 5.0$

F.3 Tax Design for Macroeconomic Models

F.3.1 Implementation Details

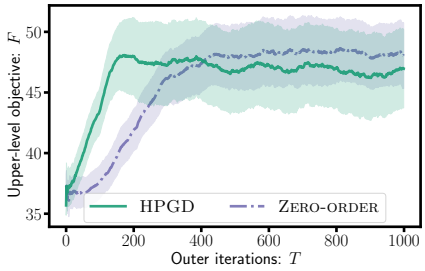
In our experiments, we use $\sigma(s) = \max(0, \log(s/20 + 1))$ and $\omega(s) = \min(s, 0)$ and select the hyperparameters as $\theta = 0.1, \phi = 5.0, \varsigma = 5.0, w = 1.0$ and $\lambda = 0.3$. We use 3 products in the simulated economy with unit prices and assume two equal-sized socio-economic groups with preferences $\alpha = (0.6, 0.3, 0.1)$ and $(0.1, 0.7, 0.2)$. Assets are initialized as $s_0 \sim N(0, \sqrt{2})$ and each episode is truncated after 200 steps. Both chosen consumption levels and hours worked are discretized with ranges $[0, 5]$ and $[0, 8]$ and cardinality 5 and 10, respectively. All tax rates are initialized as 0.3 and optimized over the continuous domain $[0.0, 2.0]$.

F.3.2 Effects of lower-level regularization

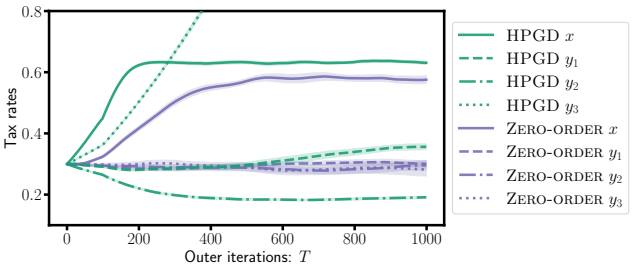
Figure 20 and Figure 21 shows the results for $\lambda = 0.3$ and $\lambda = 0.1$, respectively. Changing the regularization has small effects on the final results and only marginally change the speed of convergence, especially for the Zero-Order algorithm.

F.4 Computational Costs

We ran our experiments on a shared cluster equipped with various NVIDIA GPUs and AMD EPYC CPUs. Our default configuration for all experiments was a single GPU with 24 GB of memory, 16 CPU cores, and 4 GB of RAM per CPU core. For all parameter configurations reported in Table 3, the total runtime of the experiments for HPGD, AMD, and Zero-Order were 17, 40, and 2 hours, respectively, totaling 59 hours. Our total computational costs including the intermediate experiments are estimated to be 2-3 times more.

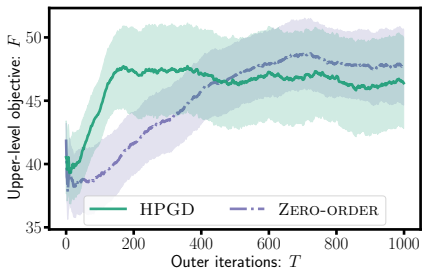


(a) Performance on the Tax Design problem

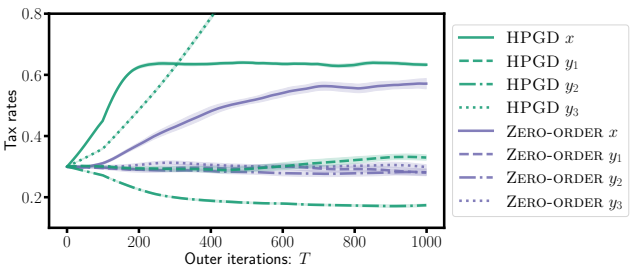


(b) Tax rates over the outer iterations.

Figure 20: Results with $\lambda = 0.3$



(a) Performance on the Tax Design problem



(b) Tax rates over the outer iterations.

Figure 21: Results with $\lambda = 0.1$

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our Abstract gives a short overview of our main contributions. Moreover in the Introduction (Section 1), we state the contributions clearly with bullet points and point to the relevant sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: For all proofs we clearly state the assumptions (see Assumption 3.1 and Assumption 3.2) we need to make and write the main limitations multiple times in the text. For the experiments we clearly point to the sensitivity of the regularization parameters.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer:[Yes]

Justification: Our Appendix E contains all proofs to all our statements. For our main theorem (Theorem 3) we further give a proof sketch in the main paper to help with intuition. Our Assumptions are clearly stated.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We describe the main experimental setting in Section 5 while provide details explanation in Appendix F.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide all code related to the research in this paper at https://github.com/lasgroup/HPGD.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All experimental details including hyperparameter selection is described in Appendix F.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the average over 10 random seeds and the corresponding standard errors for the experimental results reported in Table 3 and on Figure 1b.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report our used resources in Appendix F.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The authors have reviewed the NeurIPS Code of Ethics and conducted the research following the guidelines.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our work has a theoretical focus without an immediate impact on society.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: To our best knowledge there do not exist such risks for our work.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all codes and models used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We neither have crowdsourcing experiments nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:We neither have crowdsourcing experiments nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.