
Bilevel Reinforcement Learning via the Development of Hyper-gradient without Lower-Level Convexity

Yan Yang^{1,2} Bin Gao¹ Ya-xiang Yuan¹

¹ Academy of Mathematics and Systems Science, Chinese Academy of Sciences

² University of Chinese Academy of Sciences

Abstract

Bilevel reinforcement learning (RL), which features intertwined two-level problems, has attracted growing interest recently. The inherent non-convexity of the lower-level RL problem is, however, to be an impediment to developing bilevel optimization methods. By employing the fixed point equation associated with the regularized RL, we characterize the hyper-gradient via fully first-order information, thus circumventing the assumption of lower-level convexity. This, remarkably, distinguishes our development of hyper-gradient from the general AID-based bilevel frameworks since we take advantage of the specific structure of RL problems. Moreover, we design both model-based and model-free bilevel reinforcement learning algorithms, facilitated by access to the fully first-order hyper-gradient. Both algorithms enjoy the convergence rate $\mathcal{O}(\epsilon^{-1})$. To extend the applicability, a stochastic version of the model-free algorithm is proposed, along with results on its iteration and sample complexity. In addition, numerical experiments demonstrate that the hyper-gradient indeed serves as an integration of exploitation and exploration.

1 INTRODUCTION

Bilevel optimization, aiming to solve problems with a hierarchical structure, achieves success in a wide range of machine learning applications, e.g., hyper-parameter optimization (Franceschi et al., 2017, 2018; Ye et al.,

2023), meta-learning (Bertinetto et al., 2019), computer vision (Liu et al., 2021a), neural architecture search (Liu et al., 2019; Wang et al., 2022), adversarial training (Wang et al., 2021), reinforcement learning (Chakraborty et al., 2024; Shen et al., 2024; Thoma et al., 2024), data poisoning (Liu et al., 2024b). To address the bilevel optimization problems, a line of research has emerged recently (Ghadimi and Wang, 2018; Liu et al., 2020, 2021b; Ji et al., 2022; Hu et al., 2023; Kwon et al., 2023; Shen and Chen, 2023; Hao et al., 2024; Yao et al., 2024). Generally, the upper-level (resp. the lower-level) problem optimizes the decision taken by the leader (resp. the follower), which exhibits potential for handling complicated decision-making processes such as Markov decision processes (MDPs).

Reinforcement learning (RL) (Sutton and Barto, 2018; Szepesvári, 2022) serves as an effective way of learning to make sequential decisions in MDPs, and has seen plenty of applications (Silver et al., 2017; Berner et al., 2019; Mirhoseini et al., 2021; Miki et al., 2022; Sun et al., 2024). The central task of RL is to find the optimal policy that maximizes the expected cumulative rewards in an MDP. Bilevel RL enriches the framework of RL by considering a two-level problem: the follower solves a standard RL problem within an environment parameterized by the decision variable taken by the leader; meanwhile, the leader optimizes the decision variable based on the response policy from the lower level. Recently, bilevel RL has gained increasing attention in practice, including RL from human feedback, (Christiano et al., 2017), inverse RL (Brown et al., 2019), and reward shaping (Hu et al., 2020).

In this paper, we focus on the bilevel reinforcement learning problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \phi(x) := f(x, \pi^*(x)) \\ \text{s. t.} \quad & \pi^*(x) \in \arg \min_{\pi \in \Pi} g(x, \pi), \end{aligned} \tag{1}$$

where Π is the policy set of interest, the upper-level function f is defined on $\mathbb{R}^n \times \Pi$, the univariate function ϕ is called the *hyper-objective*, and the gradient

Table 1: Comparison among bilevel reinforcement learning algorithms.

Algorithm	Deter. or Stoch.	Conv. Rate	Inner Iter.	Oracle
PARL (Chakraborty et al., 2024)	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\log \epsilon^{-1})$	1st+2nd
PBRL (Shen et al., 2024)	Deter.	$\mathcal{O}(\lambda\epsilon^{-1})$	$\mathcal{O}(\log \lambda^2 \epsilon^{-1})$	1st
HPGD (Thoma et al., 2024)	Stoch.	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\log \epsilon^{-1})$	1st
M-SoBiRL (this work)	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(1)$	1st
SoBiRL (this work)	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\log \epsilon^{-1})$	1st
Stoc-SoBiRL (this work)	Stoch.	$\tilde{\mathcal{O}}(\epsilon^{-1.5})$	$\mathcal{O}(\log \epsilon^{-1})$	1st

of $\phi(x)$ is referred to as the *hyper-gradient* (Pedregosa, 2016; Grazzi et al., 2020; Yang et al., 2023; Chen et al., 2024) if it exists. The approximate implicit differentiation (AID) based method which resorts to the hyper-gradient has become flourishing recently (Ghadimi and Wang, 2018; Ji et al., 2021; Arbel and Mairal, 2022; Dagréou et al., 2022; Liu et al., 2023; Hu et al., 2024; Yang et al., 2025). Specifically, in each *outer iteration* of the AID-based method, one implements an inexact hyper-gradient descent step $x_{k+1} = x_k - \beta \tilde{\nabla} \phi(x_k)$, where the estimator $\tilde{\nabla} \phi(x_k)$ of the hyper-gradient is obtained with the help of a few *inner iterations*. We consider the extension of AID-based methods to the bilevel reinforcement learning problem (1). Note that AID-based methods depend on the lower-level strong convexity (Khanduri et al., 2021; Huang et al., 2022; Li et al., 2022) or uniform Polyak–Łojasiewicz (PL) condition (Huang, 2023; Chakraborty et al., 2024) to ensure the existence of the hyper-gradient. However, the lower-level problem in (1)—always an RL problem—is inherently non-convex even with strongly-convex regularization (Agarwal et al., 2020; Lan, 2023), and only the non-uniform PL condition has been established (Mei et al., 2020), which renders $\nabla \phi$ ambiguous, as stated in (Shen et al., 2024).

Recently, Chen et al. (2022a) assumed the convexity of the hyper-objective and proposed a deterministic algorithm for the bilevel RL problem. Chakraborty et al. (2024) presented an AID-based framework, assuming that the lower-level problem satisfies the uniform PL condition and the Hessian non-singularity condition. Shen et al. (2024) proposed a penalty-based bilevel RL algorithm to bypass the requirement of lower-level convexity by constructing two penalty functions. The convergence rate relies on the penalty parameter λ that is at least the order of $\epsilon^{-0.5}$. Thoma et al. (2024) designed a stochastic bilevel RL method, achieving the convergence rate $\mathcal{O}(\epsilon^{-2})$. In this paper, we develop model-based and model-free bilevel RL algorithms and extend the model-free algorithm to stochastic settings, all of which are provable and exhibit an enhanced convergence rate, without the lower-lower convex assumption; see Table 1 for a detailed comparison with

the existing works, where the notation $\tilde{\mathcal{O}}(\cdot)$ hides logarithmic terms of ϵ^{-1} .

Contributions The main contributions are summarized as follows.

Firstly, we characterize the hyper-gradient of bilevel RL problem via fully first-order information and unveil its properties by investigating the fixed point equation associated with the entropy-regularized RL problem (Nachum et al., 2017; Geist et al., 2019; Yang et al., 2019), which extends the spirits in (Christianson, 1994; Grazzi et al., 2020, 2021, 2023).

Secondly, understanding the hyper-gradient enables us to construct its estimators, upon which we devise a model-based bilevel RL algorithm, M-SoBiRL, together with a model-free version, SoBiRL. For broader applicability, we extend SoBiRL in stochastic settings by designing a sampling scheme to estimate the hyper-gradient and a framework Stoc-SoBiRL aided with a momentum technique to accommodate the stochastic hyper-gradient estimator. Specifically, the implementation only requires first-order oracles, which circumvents complicated second-order queries in general AID-based bilevel methods.

Finally, we offer an analysis to illustrate the efficiency of amortizing the hyper-gradient approximation through outer iterations in M-SoBiRL, i.e., it enjoys the convergence rate $\mathcal{O}(\epsilon^{-1})$ with the inner iteration number $N = \mathcal{O}(1)$ independent of the solution accuracy ϵ . In the model-free scenario, an enhanced convergence property is also established. Moreover, investigating the statistical properties of Stoc-SoBiRL, we show its iteration complexity $\tilde{\mathcal{O}}(\epsilon^{-1.5})$ and sample complexity $\tilde{\mathcal{O}}(\epsilon^{-3.5})$; refer to Table 1 for a detailed comparison. To the best of our knowledge, it is the first sample complexity result established for bilevel RL problems. In addition, a synthetic experiment verifies the convergence of M-SoBiRL, and the favorable performance of the proposed SoBiRL is validated on Atari games and Mujoco simulations, which implies that the hyper-gradient is an aggregation of exploitation and exploration.

2 RELATED WORK

The introduction to related bilevel optimization methods can be found in Appendix A.

Entropy-regularized Reinforcement Learning Entropy regularization is commonly considered in the RL community. Specifically, the goal of entropy-regularized RL is to maximize the expected reward augmented with the policy entropy, thereby boosting both task success and behavior stochasticity. It facilitates exploration and robustness (Ziebart, 2010; Haarnoja et al., 2017, 2018a; Lan, 2021), smoothens the optimization landscape (Ahmed et al., 2019), and enhances the convergence property (Mnih et al., 2016; Cen et al., 2022; Zhan et al., 2023; Lan, 2023; Yang et al., 2023; Li et al., 2024b). Moreover, the policy optimality condition under entropy-regularized setting is equivalent to the *softmax temporal value consistency* in (Nachum et al., 2017). Therefore, the term *soft* is prefixed to quantities in this scenario, also resonating with the concept in (Sutton and Barto, 2018), where “soft” means that the policy ensures positive probabilities across all state-action pairs.

Bilevel Reinforcement Learning Extensive reinforcement learning applications hold the bilevel structure, e.g., reward shaping (Sorg et al., 2010; Zheng et al., 2018; Hu et al., 2020; Devidze et al., 2022; Gupta et al., 2023), offline reward correction (Li et al., 2023), preference-based RL (Christiano et al., 2017; Lee et al., 2021; Saha et al., 2023; Shen et al., 2024), apprenticeship learning (Arora and Doshi, 2021), Stackleberg games (Fiez et al., 2020; Song et al., 2023; Shen et al., 2024), etc. In terms of provable bilevel reinforcement learning frameworks, Chakraborty et al. (2024) proposed a policy alignment algorithm demonstrating performance improvements, Shen et al. (2024) designed a penalty-based method, and Thoma et al. (2024) investigated the problem in the stochastic setting. However, the research related to the hyper-gradient in bilevel RL remains constrained.

3 PROBLEM FORMULATION

In this section, we will introduce the setting of entropy-regularized Markov decision processes (Williams and Peng, 1991; Ziebart, 2010; Nachum et al., 2017; Haarnoja et al., 2017), and the formulation of bilevel reinforcement learning. Moreover, two specific instances of bilevel RL are introduced. The notation is listed in Appendix B.

3.1 Entropy-regularized MDPs

Discounted Infinite-horizon MDPs An MDP is characterized by a tuple $\mathcal{M}_\tau = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \tau)$ with \mathcal{S} and \mathcal{A} serving as the state space and the action space, respectively. In this paper, we restrict the focus to the tabular setting, where both \mathcal{S} and \mathcal{A} are finite, i.e., $|\mathcal{S}| < +\infty$ and $|\mathcal{A}| < +\infty$. Furthermore, $P \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times|\mathcal{S}|}$ is the transition matrix with $P_{sas'} = P(s'|s, a)$ representing the transition probability from state s to state s' under action a . The vector $r \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ specifies the reward r_{sa} received when action a is carried out at state s . Note that $\gamma \in (0, 1)$ is the discount factor, adjusting the importance of immediate versus future reward, and the *temperature parameter* $\tau \geq 0$ balances regularization and reward.

Soft Value Function and Q-function A policy $\pi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ provides an action selection rule, that is, for any $s \in \mathcal{S}$, $a \in \mathcal{A}$, $\pi_{sa} = \pi(a|s)$ is the probability of performing action a at state s , and we denote $\pi_s \in \mathbb{R}^{|\mathcal{A}|}$ to be the corresponding distribution over \mathcal{A} at state s , which means π_s belongs to the probability simplex $\Delta(\mathcal{A}) := \{x \in \mathbb{R}^{|\mathcal{A}|} : x_i \geq 0, \sum_{i=1}^{|\mathcal{A}|} x_i = 1\}$. Collecting π_s over $s \in \mathcal{S}$ defines $\Delta^{|\mathcal{S}|}(\mathcal{A}) := \{\pi = (\pi_s)_{s \in \mathcal{S}} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} : \pi_s \in \Delta(\mathcal{A}) \text{ for } s \in \mathcal{S}\}$ as the feasible set of policies. For simplicity, the above two sets are abbreviated as Δ and $\Delta^{|\mathcal{S}|}$, respectively. A policy π induces a $P^\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$, measuring the transition probability from one state to another, i.e., $P_{ss'}^\pi = \sum_a \pi_{sa} P_{sas'}$. To promote stochasticity and discourage premature convergence to sub-optimal policies (Ahmed et al., 2019), it commonly resorts to the entropy function $h(\cdot)$,

$$h(\pi_s) := - \sum_{a \in \mathcal{A}} \pi_{sa} \log \pi_{sa}, \quad \text{for } \pi_s \in \mathbb{R}_+^{|\mathcal{A}|} \cap \Delta,$$

where $\mathbb{R}_+^m := \{z \in \mathbb{R}^m : z_i > 0\}$. Consequently, the *soft value function* $V^\pi \in \mathbb{R}^{|\mathcal{S}|}$ represents the expected discounted reward augmented with policy entropy, i.e.,

$$V_s^\pi := \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t (r_{s_t a_t} + \tau h(\pi_{s_t})) \mid s_0 = s, \pi \right],$$

for $s \in \mathcal{S}$, where the expectation is taken over the trajectory $(s_0, a_0 \sim \pi(\cdot|s_0), s_1 \sim P(\cdot|s_0, a_0), \dots)$. Given the initial state distribution $\rho \in \mathbb{R}_+^{|\mathcal{S}|} \cap \Delta(\mathcal{S})$, the objective of RL is to find the optimal policy $\pi \in \Delta^{|\mathcal{S}|}$ that solves the problem:

$$\max_{\pi \in \Delta^{|\mathcal{S}|}} V^\pi(\rho) := \mathbb{E}_{s \sim \rho} [V_s^\pi]. \quad (2)$$

The soft Q-function $Q^\pi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ associated with policy π , couples with V^π in the following fashion,

$$\begin{aligned} \forall (s, a) \in \mathcal{S} \times \mathcal{A} : Q_{sa}^\pi &= r_{sa} + \gamma \mathbb{E}_{s' \sim P(\cdot|s, a)} [V_{s'}^\pi], \\ \forall s \in \mathcal{S} : V_s^\pi &= \mathbb{E}_{a \sim \pi(\cdot|s)} [-\tau \log \pi(a|s) + Q_{sa}^\pi]. \end{aligned} \quad (3)$$

It is worth noting that the optimization problem (2) admits a unique *optimal soft policy* π^* independent of ρ (Nachum et al., 2017). The corresponding *optimal soft value function* (resp. *optimal soft Q-function*) is denoted by $V^* := V^{\pi^*}$ (resp. $Q^* := Q^{\pi^*}$). If necessary, we add a subscript to clarify the environment in which the policy is evaluated, e.g., $V_{\mathcal{M}_\tau}^\pi$ and $Q_{\mathcal{M}_\tau}^\pi$.

3.2 Bilevel RL Formulation

The single-level reinforcement learning task in Section 3.1 trains an agent with fixed rewards. In contrast, this work considers the scenario where the reward function is parameterized by a decision variable, a task dubbed “policy alignment” in (Chakraborty et al., 2024). Generally, we parameterize the reward $r(x) \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ with $x \in \mathbb{R}^n$, which results in a parameterized MDP $\mathcal{M}_\tau(x) = (\mathcal{S}, \mathcal{A}, P, r(x), \gamma, \tau)$. In parallel with Section 3.1, given $\mathcal{M}_\tau(x)$, we define the soft value function $V^\pi(x)$, the soft Q-function $Q^\pi(x)$, the objective $V_{\mathcal{M}_\tau(x)}^\pi(\rho)$, the optimal soft policy $\pi^*(x)$, the optimal soft value-function $V^*(x)$, and the optimal soft Q-function $Q^*(x)$, associated with x .

Consequently, we formulate the bilevel reinforcement learning problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \phi(x) := f(x, \pi^*(x)) \\ \text{s. t.} \quad & \pi^*(x) = \arg \min_{\pi \in \Delta^{|\mathcal{A}|}} -V_{\mathcal{M}_\tau(x)}^\pi(\rho), \end{aligned} \quad (4)$$

where the upper-level f is defined on $\mathbb{R}^n \times \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$.

3.3 Applications of Bilevel RL

We introduce two examples unified in the formulation (4). To make a distinction between environments at two levels, we denote the upper-level MDP by $\bar{\mathcal{M}}_\tau = \{\mathcal{S}, \mathcal{A}, \bar{P}, \bar{r}, \bar{\gamma}, \bar{\tau}\}$ with an initial state distribution $\bar{\rho}$, and the lower-level MDP by $\mathcal{M}_\tau(x)$ with ρ . More applications can be found in Appendix K.

Reward Shaping In reinforcement learning, the reward function acts as the guiding signal to motivate agents to achieve specified goals. However, in many cases, the rewards are sparse, which impedes the policy learning, or are partially incorrect, which leads to inaccurate policies. To this end, from the perspective of bilevel RL, it is advisable to shape an auxiliary reward function $r(x)$ at the lower level for efficient agent training, while maintaining the original environment at the upper level to align with the initial task evaluation (Hu et al., 2020), which is established as

$$\min_{x \in \mathbb{R}^n} -V_{\bar{\mathcal{M}}_\tau}^{\pi^*(x)}(\bar{\rho}), \quad \text{s. t. } \pi^*(x) = \arg \min_{\pi \in \Delta^{|\mathcal{A}|}} -V_{\mathcal{M}_\tau(x)}^\pi(\rho).$$

Reinforcement Learning from Human Feedback (RLHF) The target of RLHF is to learn the intrinsic

reward function that incorporates expert knowledge, from simple labels only containing human preferences. Drawing from the original framework (Christiano et al., 2017), Chakraborty et al. (2024) and Shen et al. (2024) have formulated it in a bilevel form, which optimizes a policy under $r(x)$ at the lower level, and adjusts x to align the preference predicted by the reward model $r(x)$ with the true labels at the upper level.

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \mathbb{E}_{y, d_1, d_2 \sim \rho(d; \pi^*(x))} [l_h(d_1, d_2, y; x)] \\ \text{s. t.} \quad & \pi^*(x) = \arg \min_{\pi \in \Delta^{|\mathcal{A}|}} -V_{\mathcal{M}_\tau(x)}^\pi(\rho), \end{aligned}$$

where each trajectory $d_i = \{(s_h^i, a_h^i)\}_{h=0}^{H-1}$ ($i = 1, 2$) is sampled from the distribution $\rho(d; \pi^*(x))$ generated by the policy $\pi^*(x)$ in the upper-level $\bar{\mathcal{M}}_\tau$, i.e.,

$$P(d_i) = \bar{\rho}(s_0^i) [\prod_{h=0}^{H-2} \pi^*(x)(a_h^i | s_h^i) \bar{P}(s_{h+1}^i | s_h^i, a_h^i)] \times \pi^*(x)(a_{H-1}^i | s_{H-1}^i),$$

and the preference label $y \in \{0, 1\}$, indicating preference for d_1 over d_2 , obeys human feedback distribution $y \sim D_{\text{human}}(y|d_1, d_2)$. Moreover, $l_h(d_1, d_2, y; x) = -y \log P(d_1 \succ d_2; x) - (1-y) \log P(d_2 \succ d_1; x)$, with the preference probability $P(d_1 \succ d_2; x)$ built by the Bradley–Terry model; see (102) in the appendix.

4 MODEL-BASED SOFT BILEVEL REINFORCEMENT LEARNING

Addressing the bilevel RL problem (4) is challenging from two aspects: 1) analyzing the properties of $\pi^*(x)$, which has not been well studied but plays a crucial role in AID-based methods; 2) characterizing the hypergradient $\nabla \phi(x)$ which remains unclear. To this end, we take advantage of the specific structure of entropy-regularized RL to investigate $\pi^*(x)$ and identify $\nabla \phi(x)$. As a result, we propose a model-based algorithm to solve the bilevel RL problem (4).

Recall that the optimal soft quantities adhere to the following softmax temporal value consistency conditions (Nachum et al., 2017): $\forall (s, a) \in \mathcal{S} \times \mathcal{A}$,

$$\pi_{sa}^*(x) = \exp\{Q_{sa}^*(x)/\tau\} / \left(\sum_{a'} \exp(Q_{sa'}^*(x)/\tau) \right), \quad (5)$$

$$V_s^*(x) = \tau \log \left(\sum_a \exp \left(\frac{r_{sa}(x) + \gamma \sum_{s'} P_{sas'} V_{s'}^*(x)}{\tau} \right) \right) \quad (6)$$

$$= \tau \log \left(\sum_a \exp(Q_{sa}^*(x)/\tau) \right). \quad (7)$$

By differentiating (5) and incorporating (3), we assemble $\{\nabla \pi_{sa}^*(x)\}$ in matrix form and obtain

$$\nabla \pi^*(x) = \tau^{-1} \text{diag}(\pi^*(x)) (\nabla r(x) - U \nabla V^*(x)) \quad (8)$$

with $U \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|}$ defined by $U_{sas'} := 1 - \gamma P_{sas'}$ for $s = s'$ and $U_{sas'} := -\gamma P_{sas'}$, otherwise. Hence, differentiating $\pi^*(x)$ boils down to considering the *implicit differentiation* $\nabla V^*(x)$. In view of (6), we notice that $V^*(x)$ is a fixed point related to the mapping φ ,

$$\varphi(x, v) : \mathbb{R}^n \times \mathbb{R}^{|\mathcal{S}|} \longrightarrow \mathbb{R}^{|\mathcal{S}|}$$

$$(x, v) \longmapsto \tau \log \left(\exp \left(\frac{r(x) + \gamma Pv}{\tau} \right) \cdot \mathbf{1} \right),$$

where $\log(\cdot)$ and $\exp(\cdot)$ are element-wise operations, and $\mathbf{1} \in \mathbb{R}^{|\mathcal{S}|}$ is an all-ones vector. The structure of $\varphi(\cdot, \cdot)$, in consequence, can facilitate the characterization of $\nabla V^*(x)$, as outlined in the following proposition.

Proposition 4.1. *For any $x \in \mathbb{R}^n$, $\varphi(x, \cdot)$ is a contraction mapping, i.e., $\|\nabla_v \varphi(x, v)\|_\infty = \gamma < 1$, and the matrix $I - \nabla_v \varphi(x, v)$ is invertible. Consequently, $V^*(x)$ is the unique fixed point of $\varphi(x, \cdot)$, with a well-defined derivative $\nabla V^*(x)$, given by*

$$\nabla V^*(x) = (I - \nabla_v \varphi(x, V^*(x)))^{-1} \nabla_x \varphi(x, V^*(x)).$$

Additionally, $\nabla_v \varphi(x, V^*(x))$ coincides with the γ -scaled transition matrix induced by the optimal soft policy $\pi^*(x)$, i.e., $\nabla_v \varphi(x, V^*(x)) = \gamma P^{\pi^*(x)}$.

Combining the above proposition with (8), we identify the hyper-gradient $\nabla \phi(x)$ to the problem (4); its expression is delayed in (24) of Appendix G. In this manner, we unveil the hyper-gradient in the context of bilevel RL, by harnessing the softmax temporal value consistency and derivatives of a fixed-point equation.

In light of the characterization of $\nabla \phi$, a model-based soft bilevel reinforcement learning algorithm, called M-SoBiRL, is proposed in Algorithm 2. In summary, the k -th outer iteration performs an inexact hyper-gradient descent step on x_k , with the aid of auxiliary iterates (π_k, Q_k, V_k, w_k) . Specifically, given x_{k+1} , the inner iterations aim to (approximately) solve the lower-level entropy-regularized MDP problem. To this end, the *soft policy iteration* studied by Haarnoja et al. (2018a); Cen et al. (2022) inspires us to define the following soft Bellman optimality operator $\mathcal{T}_{\mathcal{M}_\tau(x)} : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \rightarrow \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ associated with $x \in \mathbb{R}^n$,

$$\mathcal{T}_{\mathcal{M}_\tau(x)}(Q)(s, a) := r_{sa}(x) + \gamma \mathbb{E} [\tau \log (\|\exp(Q_{s', \cdot}/\tau)\|_1)],$$

where the expectation is taken with respect to $s' \sim P(\cdot | s, a)$. Applying this operator iteratively leads to the optimal soft Q-value function, with a linear convergence rate in theory (see Appendix G.4). Therefore, N inner iterations are invoked in line 7-9 of Algorithm 2 to estimate $Q^*(x_{k+1})$, and the warm-start strategy is adopted in line 10 of Algorithm 2 to initialize the next outer iteration with historical information, where it amortizes the computation of $Q^*(x)$ through the outer iterations. Additionally, in order to recover π_{k+1}

from Q_{k+1} , we take into account the consistency condition (5) which involves the softmax mapping as follows,

$$\begin{aligned} \text{softmax} : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} &\longrightarrow \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \\ \theta &\longmapsto \left(\frac{\exp(\theta(s, a))}{\sum_{a'} \exp(\theta(s, a'))} \right)_{sa}. \end{aligned}$$

It follows from (5) that $\pi_{k+1} = \text{softmax}(Q_{k+1}/\tau)$. As for the approximation of the hyper-gradient (24) which requires an inverse matrix vector product, we denote

$$A_k = (I - \gamma P^{\pi_k})^\top, \quad b_k = U^\top \text{diag}(\pi_k) \nabla_\pi f(x_k, \pi_k),$$

and employ w_k to track the quantity $A_k^{-1} b_k$ in a similar principle of amortization. Concretely, w_k is regarded as the (approximate) solution of the least squares problem, $\min_{w \in \mathbb{R}^{|\mathcal{S}|}} \frac{1}{2} \|A_k w - b_k\|^2$, and implements one gradient descent step (line 4 of Algorithm 2) in each outer iteration. Furthermore, elements of Q_k are assembled to estimate the soft value function:

$$\forall s \in \mathcal{S} : V_k(s) = \tau \log \left(\sum_a \exp(Q_k(s, a)/\tau) \right), \quad (9)$$

which complies with the form of (7). Finally, collecting the iterates (x_k, π_k, V_k, w_k) yields the model-based hyper-gradient estimator $\widehat{\nabla} \phi$, i.e.,

$$\begin{aligned} \widehat{\nabla} \phi(x_k, \pi_k, V_k, w_k) &:= \nabla_x f(x_k, \pi_k) + \frac{1}{\tau} \nabla_r(x_k)^\top \text{diag}(\pi_k) \\ &\quad \times \nabla_\pi f(x_k, \pi_k) - \frac{1}{\tau} \nabla_x \varphi(x_k, V_k)^\top w_k. \end{aligned}$$

5 MODEL-FREE SOFT BILEVEL REINFORCEMENT LEARNING

In many real-world applications, agents lack access to the accurate model P of the environment, underscoring the need for efficient model-free algorithms (Schulman et al., 2015, 2016; Lillicrap et al., 2016; Mnih et al., 2016; Schulman et al., 2017; Haarnoja et al., 2018a). When P is unknown, two obstacles appear from the hyper-gradient computation (24). Specifically, it explicitly relies on the black-box transition matrix P to compute $\nabla V^*(x)$, i.e.,

$$\nabla V^*(x) = (I - \gamma P^{\pi^*(x)})^{-1} \nabla_x \varphi(x, V^*(x)), \quad (10)$$

and the computation involves complicated large-scale matrix multiplications. To circumvent them, we demonstrate how to derive the hyper-gradient in an expectation, which allows us to estimate $\nabla \phi(x)$ via sampling fully first-order information. Following these developments, we offer a model-free soft bilevel reinforcement learning algorithm.

In the model-free scenario, we concentrate on the upper-level function

$$f(x, \pi) = \mathbb{E}_{d_i \sim \rho(d; \pi)} [l(d_1, d_2, \dots, d_I; x)], \quad (11)$$

where each trajectory $d_i = \{(s_h^i, a_h^i)\}_{h=0}^{H-1}$ is independently sampled from the trajectory distribution $\rho(d; \pi)$ generated by the policy π in the upper-level $\bar{\mathcal{M}}_{\bar{\tau}}$, and $l(\cdot; x)$ associated with x is a function of trajectories. Notice that it incorporates the applications in Section 3.3, with $I = 1$, $H = 1$, $l = -Q_{\bar{\mathcal{M}}_{\bar{\tau}}}^{\pi^*(x)}(s_0^1, a_0^1)$ for reward shaping, and $I = 2$, finite H , $l = \mathbb{E}_y[l_h(d_1, d_2, y; x)]$ for RLHF.

We extend Proposition 4.1 to absorb the transition matrix P in (10) into an expectation, with the result that the implicit differentiations $\nabla V^*(x)$ and $\nabla Q^*(x)$ can be estimated by sampling the reward gradient under the policy $\pi^*(x)$.

Proposition 5.1. *For any $x \in \mathbb{R}^n$, $s \in \mathcal{S}$, and $a \in \mathcal{A}$,*

$$\begin{aligned}\nabla V_s^*(x) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \nabla r_{s_t, a_t}(x) \mid s_0 = s, \pi^*(x)\right], \\ \nabla Q_{sa}^*(x) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \nabla r_{s_t, a_t}(x) \mid s_0 = s, a_0 = a, \pi^*(x)\right].\end{aligned}$$

Nevertheless, it is still intractable to construct the $|\mathcal{S}| \times n$ matrix $\nabla V^*(x)$ based on the above element-wise calculation, not to mention the large-scale matrix multiplications in (24). To bypass these matrix computations, we resort to the “log probability trick” (Sutton and Barto, 2018) and the consistency condition (5). The subsequent proposition confirms that we can evaluate the hyper-gradient by interacting with the environment and collecting fully first-order information.

Proposition 5.2 (Hyper-gradient). *The hyper-gradient $\nabla \phi(x)$ of the bilevel reinforcement learning problem (4) with upper-level function (11) can be computed by*

$$\nabla \phi(x) = \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} [\nabla l(d_1, d_2, \dots, d_I; x)] \quad (12)$$

$$+ \tau^{-1} \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} \left[l(d_1, d_2, \dots, d_I; x) \right. \\ \left. \left(\sum_i \sum_h \nabla \left(Q_{s_h^i a_h^i}^*(x) - V_{s_h^i}^*(x) \right) \right) \right]. \quad (13)$$

Essentially, the first term (12) in the hyper-gradient contributes to decreasing the upper-level function, and the second term (13) aggregates the gradient information transmitted from the lower level. Assembling these two directions constructs the hyper-gradient, and updating the upper-level variable x along $-\nabla \phi(x)$ will lead to a first-order stationary point (see Theorem 7.6).

In line with the above propositions, to facilitate the k -th (inexact) hyper-gradient descent step, the construction of a hyper-gradient estimator $\tilde{\nabla} \phi(x_k, \pi_k)$ is divided into two steps: 1) evaluating the implicit dif-

ferentiations based on π_k ,

$$\tilde{\nabla} V_s(x_k, \pi_k) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \nabla r_{s_t, a_t}(x_k) \mid s_0 = s, \pi_k\right],$$

$$\tilde{\nabla} Q_{sa}(x_k, \pi_k) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \nabla r_{s_t, a_t}(x_k) \mid s_0 = s, a_0 = a, \pi_k\right];$$

2) absorbing these components into the upper-level sampling process induced by π_k ,

$$\begin{aligned}\tilde{\nabla} \phi(x_k, \pi_k) &= \mathbb{E}_{d_i \sim \rho(d; \pi_k)} [\nabla l(d_1, d_2, \dots, d_I; x_k)] \quad (14) \\ &+ \tau^{-1} \mathbb{E}_{d_i \sim \rho(d; \pi_k)} \left[l(d_1, d_2, \dots, d_I; x_k) \right. \\ &\left. \left(\sum_i \sum_h \tilde{\nabla} \left(Q_{s_h^i a_h^i}^* - V_{s_h^i}^* \right)(x_k, \pi_k) \right) \right].\end{aligned}$$

Consequently, we propose a model-free soft bilevel reinforcement learning algorithm, called SoBiRL, which is outlined in Algorithm 1. In the k -th outer iteration, we search for an approximate optimal soft policy π_k satisfying $\|\pi_k - \pi^*(x_k)\|_2^2 \leq \epsilon$, which can be achieved by executing $\mathcal{O}(\log \epsilon^{-1})$ iterations of the policy mirror descent algorithm (Lan, 2023; Zhan et al., 2023). Then, we utilize $\tilde{\nabla} \phi(x_k, \pi_k)$, whose estimation only involves first-order oracles, for an inexact hyper-gradient descent step.

Algorithm 1 Model-Free Soft Bilevel Reinforcement Learning Algorithm (SoBiRL)

Input: iteration number K , step size β , initialization x_1, π_0 , required accuracy ϵ
 1: **for** $k = 1, \dots, K$ **do**
 2: Solve the lower-level problem with the initialization π_{k-1} to get π_k with $\|\pi_k - \pi_k^*\|_2^2 \leq \epsilon$
 3: Compute $\tilde{\nabla} \phi(x_k, \pi_k)$
 4: Implement an inexact hyper-gradient descent step $x_{k+1} = x_k - \beta \tilde{\nabla} \phi(x_k, \pi_k)$
 5: **end for**

Output: (x_{K+1}, π_{K+1})

6 THE STOCHASTIC EXTENSION

The preceding sections establish bilevel methods through the lens of deterministic optimization. However, in practical implementations of reinforcement learning, quantities are typically approximated via sampling data, e.g., $\tilde{\nabla} \phi(x_k, \pi_k)$ defined in an expected form. As a result, the sampling process introduces stochasticity to the algorithm. Therefore, to enrich the developed framework SoBiRL both practically and theoretically, it is essential to extend it to stochastic settings. In detail, we approach this in two steps. (1) Designing a scheme to estimate the hyper-gradient $\nabla \phi$ by sampling trajectories and then characterizing the resulting

stochasticity, i.e., the bias and variance. (2) Replacing the hyper-gradient in SoBiRL with its stochastic counterpart and incorporating a momentum technique to accommodate the stochasticity, leading to our stochastic variant, Stoc-SoBiRL.

In the stochastic scenario, we still focus on the problem formulation (4) with the upper-level function (11) as discussed in Section 5. Denote the trajectory tuple (d_1, d_2, \dots, d_I) as \mathbf{d} for simplicity. In the k -th outer iteration, to estimate $\nabla\phi$, an expectation with respect to the random variable \mathbf{d} , we can generate M independent tuples $\mathbf{d}^m := (d_1^m, d_2^m, \dots, d_I^m)$ for $m = 1, 2, \dots, M$, evaluate corresponding quantities along each \mathbf{d}^m and average them. To this end, it is necessary to tackle the terms ∇Q^* and ∇V^* in (13). Assuming access to a generative model (Kearns and Singh, 1998; Li et al., 2024a), from any initial state-action pair (s, a) , we sample J independent trajectories of length T by implementing π_k , i.e., for $j = 1, 2, \dots, J$,

$$\xi_k^j(s, a) := \{(s_0^j = s, a_0^j = a); (s_1^j, a_1^j), \dots, (s_{T-1}^j, a_{T-1}^j)\}.$$

Collecting all these random variables yields $\boldsymbol{\xi}_k := \{\xi_k^j(s, a), j = 1, \dots, J, s \in \mathcal{S}, a \in \mathcal{A}\}$, and the estimator for ∇Q^* can be constructed as follows.

$$\bar{\nabla}Q_{sa}^{\boldsymbol{\xi}_k}(x_k, \pi_k) := \frac{1}{J} \sum_{j=1}^J \sum_{t=0}^{T-1} \gamma^t \nabla r_{s_t^j a_t^j}(x_k).$$

The random variable $\boldsymbol{\zeta}_k$ and the associated estimator $\bar{\nabla}V^{\boldsymbol{\zeta}_k}$ for ∇V^* are constructed similarly. Consequently, denoting $\mathbf{D}_k := (\mathbf{d}_k^1, \mathbf{d}_k^2, \dots, \mathbf{d}_k^M)$, we obtain the stochastic hyper-gradient:

$$\begin{aligned} \bar{\nabla}\phi(\mathbf{D}_k, \boldsymbol{\xi}_k, \boldsymbol{\zeta}_k; x_k, \pi_k) &= \frac{1}{M} \sum_{m=1}^M \nabla l(\mathbf{d}_k^m; x_k) + \frac{1}{\tau M} \\ &\times \sum_{m=1}^M l(\mathbf{d}_k^m; x_k) \sum_i \sum_h \bar{\nabla} \left(Q_{s_h^{m,i} a_h^{m,i}}^{\boldsymbol{\xi}_k} - V_{s_h^{m,i}}^{\boldsymbol{\zeta}_k} \right)(x_k, \pi_k), \end{aligned}$$

which is abbreviated as $\bar{\nabla}\phi_k$ in the following discussion. The sampling scheme is outlined in Algorithm 3.

Momentum techniques are known to be beneficial for reducing variance and accelerating algorithms (Cutkosky and Orabona, 2019). To this end, we maintain a momentum-instructed h_k in the k -th outer iteration,

$$h_k = \bar{\nabla}\phi_k + (1 - \mu_k)(h_{k-1} - \bar{\nabla}\phi(\mathbf{D}_k, \boldsymbol{\xi}_k, \boldsymbol{\zeta}_k; x_{k-1}, \pi_{k-1})),$$

which tracks $\bar{\nabla}\phi_k$ via current and historical hyper-gradient estimates. Using h_k to update the upper-level x_k , we obtain the stochastic Algorithm 4, Stoc-SoBiRL.

7 THEORETICAL ANALYSIS

In this section, we prove global convergence and give the iteration and sample complexity for the proposed

algorithms. The analysis requires only the regularity conditions—specifically, boundedness and Lipschitz continuity—of the first-order information. This distinguishes from general AID-based methods, which require additional smoothness assumptions regarding second-order derivatives (Ghadimi and Wang, 2018; Chen et al., 2021; Khanduri et al., 2021; Dagréou et al., 2022; Ji et al., 2022). The properties of φ , $\pi^*(x)$, and $V^*(x)$ are outlined in Appendix F. Moreover, it is worth emphasizing that the analysis for M-SoBiRL is different from that for SoBiRL. Thus, we organize their properties and the detailed proofs in Appendix G and Appendix H. The analysis for Stoc-SoBiRL is referred to Appendix I.

Assumption 7.1. In the model-based scenario (4), f is continuously differentiable. The gradient $\nabla f(x, \pi)$ is L_f -Lipschitz continuous, i.e., for any $x_1, x_2 \in \mathbb{R}^n, \pi_1, \pi_2 \in \Delta^{|\mathcal{S}|}$, $\|\nabla f(x_1, \pi_1) - \nabla f(x_2, \pi_2)\|_2 \leq L_f(\|x_1 - x_2\|_2 + \|\pi_1 - \pi_2\|_\infty)$. Additionally, the boundedness condition holds, $\|\nabla f(x, \pi^*(x))\|_2 \leq C_{f\pi}$.

Assumption 7.2. In the model-free scenario with upper-level function (11), denote $\mathbf{d} = (d_1, d_2, \dots, d_I)$. $l(\mathbf{d}; x)$ parameterized by x is a function of I trajectories, each consisting of H steps. It is bounded, i.e., $|l(\mathbf{d}; x)| \leq C_l$, and is continuously differentiable with respect to x . $l(\mathbf{d}; x)$ is L_l -Lipschitz continuous, i.e., for any $x_1, x_2 \in \mathbb{R}^n$ and \mathbf{d} , $\|l(\mathbf{d}; x_1) - l(\mathbf{d}; x_2)\|_2 \leq L_l\|x_1 - x_2\|_2$. Moreover, $\nabla l(\mathbf{d}; x)$ is L_{l1} -Lipschitz continuous, i.e., for any $x_1, x_2 \in \mathbb{R}^n$ and \mathbf{d} , $\|\nabla l(\mathbf{d}; x_1) - \nabla l(\mathbf{d}; x_2)\|_2 \leq L_{l1}\|x_1 - x_2\|_2$.

Assumptions 7.1 and 7.2 specify the requirements for the upper-level problems in model-based and model-free scenarios, respectively. Generally, the boundedness and Lipschitz continuity assumptions related to f are standard in bilevel optimization (Ghadimi and Wang, 2018; Ji et al., 2021; Arbel and Mairal, 2022; Chen et al., 2022b; Li et al., 2022).

Assumption 7.3. The reward is bounded by C_r , i.e., $|r_{sa}(x)| \leq C_r$, and the gradient is bounded by C_{rx} , i.e., $\|\nabla r_{sa}(x)\|_2 \leq C_{rx}$. Additionally, $r_{sa}(x)$ is L_r -Lipschitz smooth, i.e., for any $x_1, x_2 \in \mathbb{R}^n$ and $(s, a) \in \mathcal{S} \times \mathcal{A}$, $\|\nabla r_{sa}(x_1) - \nabla r_{sa}(x_2)\|_2 \leq L_r\|x_1 - x_2\|_2$.

This assumption on the parameterized reward function is common in RL (Zhang et al., 2020; Lan et al., 2022; Chakraborty et al., 2024; Shen et al., 2024).

In the model-based scenario, the following theorem reveals that the proposed algorithm M-SoBiRL benefits from amortizing the hyper-gradient approximation in which it enjoys the $\mathcal{O}(\epsilon^{-1})$ convergence rate, and the inner iteration number N can be selected as a constant independent of the accuracy ϵ .

Theorem 7.4 (Model-based). *Under Assumptions 7.1 and 7.3, in M-SoBiRL, we can choose constant step*

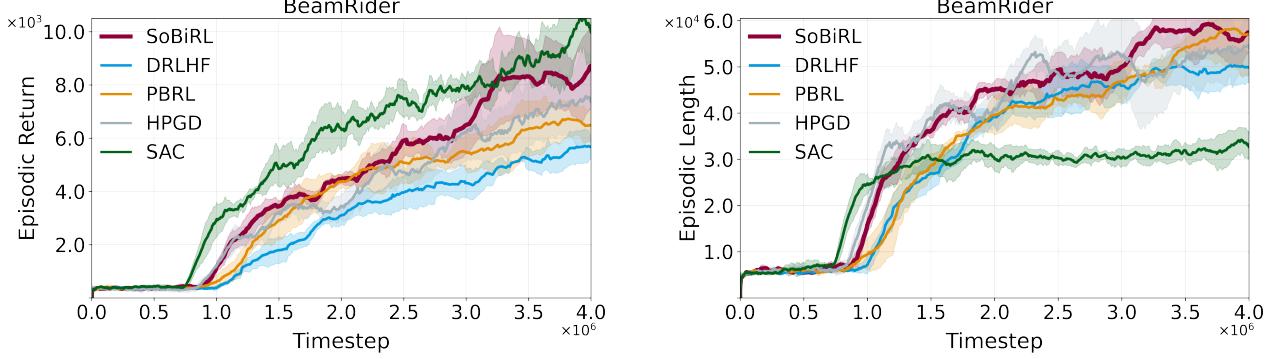


Figure 1: Comparison of algorithms on the Atari game, BeamRider, evaluated by the ground-truth reward. Each bilevel algorithm collects a total of 3000 trajectory pairs. The running average over 15 consecutive episodes is adopted for the presentation, and results are averaged over 5 seeds.

sizes β, η , and the inner iteration number $N \sim \mathcal{O}(1)$. Then the iterates $\{x_k\}$ satisfy

$$\frac{1}{K} \sum_{k=1}^K \|\nabla \phi(x_k)\|_2^2 = \mathcal{O}(K^{-1}).$$

Detailed parameter setting is referred to Theorem G.12.

Coupled with Proposition 5.2, the next proposition marks a substantial development to clarify the Lipschitz property of the hyper-gradient.

Proposition 7.5. Under Assumption 7.3, we have $\|\nabla \phi(x_1) - \nabla \phi(x_2)\|_2 \leq L_\phi \|x_1 - x_2\|_2$ for any $x_1, x_2 \in \mathbb{R}^n$, with L_ϕ specified in Proposition H.6.

Theorem 7.6 (Model-free). Under Assumptions 7.2 and 7.3 and given the accuracy ϵ in Algorithm 1, we can set the constant $\beta < \frac{1}{2L_\phi}$, and then the iterates $\{x_k\}$ satisfy

$$\frac{1}{K} \sum_{k=1}^K \|\nabla \phi(x_k)\|_2^2 \leq \frac{\phi(x_1) - \phi^*}{K \left(\frac{\beta}{2} - \beta^2 L_\phi \right)} + \frac{1 + 2\beta L_\phi}{1 - 2\beta L_\phi} L_\phi^2 \epsilon,$$

where ϕ^* is the minimum of $\phi(x)$, and $L_\phi, L_{\tilde{\phi}}$ are specified in Propositions H.6 and H.7, respectively.

Combined with Table 1, the convergence property in Theorem 7.6 implies that SoBiRL realizes better iteration complexity than PBRL (Shen and Chen, 2023) with the same computation cost at each outer and inner iteration. Moreover, SoBiRL attains the convergence rate of the same order as PARL (Chakraborty et al., 2024), but only employs first-order oracles and gets rid of the convexity assumption on the lower-level problem.

In the context of stochastic bilevel RL, the distribution of samples (D_k, ξ_k, ζ_k) relies on the variable π_k , which differs from related work employing momentum techniques (Cutkosky and Orabona, 2019; Khanduri et al.,

2021; Huang, 2023). To circumvent this misalignment in Stoc-SoBiRL, we give the statistical properties (bias and variance of the stochastic hyper-gradient) in Appendix I.2 and the convergence analysis in Appendix I.3.

Theorem 7.7 (Stochastic). Under Assumptions 7.2 and 7.3, we can choose appropriate sampling configurations $M \sim \mathcal{O}(K^{4/3})$, $J \sim \mathcal{O}(1)$, $T \sim \mathcal{O}(\log K)$. Then the iterates $\{x_k\}$ generated by Algorithm 4 satisfy

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} \left[\|\nabla \phi(x_k)\|_2^2 \right] = \mathcal{O} \left(\frac{\log K}{K^{2/3}} \right) = \tilde{\mathcal{O}} \left(K^{-\frac{2}{3}} \right).$$

Detailed parameter setting is given in Theorem I.7.

Enlightened by the momentum acceleration, the average norm square of the hyper-gradient achieves ϵ within $\tilde{\mathcal{O}}(\epsilon^{-1.5})$ outer iterations. Therefore, the samples required by the upper-level leader are of the order $\tilde{\mathcal{O}}(\epsilon^{-3.5})$, which can be alleviated in practice, as the leader can leverage vectorized environments (Makoviychuk et al., 2021) within each outer iteration; see Appendix L for further discussion.

8 EXPERIMENTS

In this section, we conduct RLHF experiments on Atari games and Mujoco environments to validate the efficiency of SoBiRL, and one synthetic experiment to verify the convergence of M-SoBiRL. The experiment details and parameter settings are provided in Appendix J. We have made the code available on <https://github.com/UCAS-YanYang/SoBiRL>.

In RLHF experiments, we compare SoBiRL with the bilevel algorithms DRLHF (Christiano et al., 2017), PBRL (Shen et al., 2024), HPGD Thoma et al. (2024), and a baseline algorithm SAC (Haarnoja et al., 2018a,b). All the bilevel solvers harness deep neural

Table 2: Comparison of algorithms on the Atari games and Mujoco simulations, evaluated by the ground-truth reward. Each bilevel algorithm collects a total of 3000 trajectory pairs. The results are recorded after appropriate timesteps and averaged over 5 seeds.

Algorithm	BeamRider (4M)	Seaquest (1M)	SpaceInvaders (1M)	HalfCheetah (1M)	Walker2d (1M)	Hopper (500k)
SoBiRL	8459.7 ± 1989.1	746.7 ± 16.4	595.6 ± 23.8	4283.6 ± 363.0	2996.0 ± 97.4	2812.4 ± 217.5
DRLHF	6124.5 ± 433.1	561.9 ± 38.0	483.9 ± 27.3	3466.7 ± 424.2	2389.4 ± 101.8	2455.8 ± 457.2
PBRL	6798.2 ± 784.0	651.2 ± 24.6	554.0 ± 33.4	4022.8 ± 311.8	2257.8 ± 127.9	2799.1 ± 155.5
HPGD	7503.9 ± 1863.6	728.1 ± 57.5	601.8 ± 36.0	3975.8 ± 476.1	2487.5 ± 223.8	2803.3 ± 354.5
SAC	9780.0 ± 1877.4	891.1 ± 31.3	783.8 ± 56.7	5769.9 ± 765.3	4552.5 ± 23.3	2207.6 ± 18.4

networks to predict rewards, while the baseline SAC receives ground-truth rewards for training. The results are reported in Figures 1, 2, 3, and Table 2, showing that even in the face of unknown rewards, the performance of SoBiRL is on a par with the baseline SAC. Additionally, SoBiRL achieves a higher episodic return than the compared methods within the same timesteps. Another interesting observation is that, taking BeamRider for example, compared to the ground truth, the preference prediction based on the reward model of DRLHF achieves an accuracy of approximately 85%, while it hovers around 54% with the reward model of SoBiRL. This stems from different update rules. Specifically, DRLHF alternates between learning a policy given the parameterized reward $r(x)$ and minimizing $\mathbb{E}[l_h]$ based on collecting trajectories, which decouples the task into two separate phases. It aligns preferences, but overlooks the exploration for rewards promoting better policy. Nevertheless, SoBiRL glues the two levels via the hyper-gradient $\nabla\phi$, with the first term (12) exploiting the preference information to align the reward predictor and the second term (13) exploring the implicit reward to unearth a more favorable policy. Therefore, SoBiRL produces superior results in reward prediction, although it exhibits lower accuracy in alignment of trajectory preference.

Regarding M-SoBiRL, the results of a synthetic experiment are shown in Figure 4. The curves exhibit the benign convergence property of the proposed algorithm.

Acknowledgments

Bin Gao was supported by the Young Elite Scientist Sponsorship Program by CAST. Ya-xiang Yuan was supported by the National Natural Science Foundation of China (grant No. 12288201). The authors are grateful to the Program and Area Chairs and Reviewers for their detailed and valuable comments and suggestions.

References

- Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. (2020). Optimality and approximation with policy gradient methods in Markov decision processes. In *Annual Conference on Learning Theory*, pages 64–66. PMLR.
- Ahmed, Z., Le Roux, N., Norouzi, M., and Schuurmans, D. (2019). Understanding the impact of entropy on policy optimization. In *International Conference on Machine Learning*, pages 151–160. PMLR.
- Arbel, M. and Mairal, J. (2022). Amortized implicit differentiation for stochastic bilevel optimization. In *International Conference on Learning Representations*.
- Arora, S. and Doshi, P. (2021). A survey of inverse reinforcement learning: challenges, methods and progress. *Artificial Intelligence*, 297:103500.
- Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013). The arcade learning environment: an evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279.
- Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al. (2019). Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*.
- Bertinetto, L., Henriques, J., Torr, P., and Vedaldi, A. (2019). Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*.
- Brown, D., Goo, W., Nagarajan, P., and Niekum, S. (2019). Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International Conference on Machine Learning*, pages 783–792. PMLR.
- Cen, S., Cheng, C., Chen, Y., Wei, Y., and Chi, Y. (2022). Fast global convergence of natural policy gradient methods with entropy regularization. *Operations Research*, 70(4):2563–2578.
- Chakraborty, S., Bedi, A., Koppel, A., Wang, H., Manocha, D., Wang, M., and Huang, F. (2024). PARM: a unified framework for policy alignment in

- reinforcement learning. In *International Conference on Learning Representations*.
- Chen, L., Xu, J., and Zhang, J. (2024). On finding small hyper-gradients in bilevel optimization: hardness results and improved analysis. In *Annual Conference on Learning Theory*, pages 947–980. PMLR.
- Chen, S., Yang, D., Li, J., Wang, S., Yang, Z., and Wang, Z. (2022a). Adaptive model design for Markov decision process. In *International Conference on Machine Learning*, pages 3679–3700. PMLR.
- Chen, T., Sun, Y., Xiao, Q., and Yin, W. (2022b). A single-timescale method for stochastic bilevel optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2466–2488. PMLR.
- Chen, T., Sun, Y., and Yin, W. (2021). Closing the gap: tighter analysis of alternating stochastic gradient methods for bilevel problems. In *Advances in Neural Information Processing Systems*.
- Chen, Z., Zhou, Y., Chen, R.-R., and Zou, S. (2022c). Sample and communication-efficient decentralized actor-critic algorithms with finite-time analysis. In *International Conference on Machine Learning*, pages 3794–3834. PMLR.
- Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30.
- Christianson, B. (1994). Reverse accumulation and attractive fixed points. *Optimization Methods and Software*, 3(4):311–326.
- Cutkosky, A. and Orabona, F. (2019). Momentum-based variance reduction in non-convex sgd. In *Advances in Neural Information Processing Systems*, volume 32.
- Dagréou, M., Ablin, P., Vaiter, S., and Moreau, T. (2022). A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. In *Advances in Neural Information Processing Systems*, volume 35, pages 26698–26710.
- Devidze, R., Kamalaruban, P., and Singla, A. (2022). Exploration-guided reward shaping for reinforcement learning under sparse rewards. In *Advances in Neural Information Processing Systems*, volume 35, pages 5829–5842.
- Fiez, T., Chasnov, B., and Ratliff, L. (2020). Implicit learning dynamics in Stackelberg games: equilibria characterization, convergence analysis, and empirical study. In *International Conference on Machine Learning*, pages 3133–3144. PMLR.
- Franceschi, L., Donini, M., Frasconi, P., and Pontil, M. (2017). Forward and reverse gradient-based hyperparameter optimization. In *International Conference on Machine Learning*, pages 1165–1173. PMLR.
- Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., and Pontil, M. (2018). Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1568–1577. PMLR.
- Geist, M., Scherrer, B., and Pietquin, O. (2019). A theory of regularized Markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169. PMLR.
- Ghadimi, S. and Wang, M. (2018). Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*.
- Grazzi, R., Franceschi, L., Pontil, M., and Salzo, S. (2020). On the iteration complexity of hypergradient computation. In *International Conference on Machine Learning*, pages 3748–3758. PMLR.
- Grazzi, R., Pontil, M., and Salzo, S. (2021). Convergence properties of stochastic hypergradients. In *International Conference on Artificial Intelligence and Statistics*, pages 3826–3834. PMLR.
- Grazzi, R., Pontil, M., and Salzo, S. (2023). Bilevel optimization with a lower-level contraction: optimal sample complexity without warm-start. *Journal of Machine Learning Research*, 24(167):1–37.
- Gupta, D., Chandak, Y., Jordan, S., Thomas, P. S., and C da Silva, B. (2023). Behavior alignment via reward function optimization. In *Advances in Neural Information Processing Systems*, volume 36.
- Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. (2017). Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pages 1352–1361. PMLR.
- Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018a). Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR.
- Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. (2018b). Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.
- Hao, J., Gong, X., and Liu, M. (2024). Bilevel optimization under unbounded smoothness: a new algorithm and convergence analysis. In *International Conference on Learning Representations*.
- He, Y., Hu, J., Huang, X., Lu, S., Wang, B., and Yuan, K. (2024). Distributed bilevel optimization with communication compression. In *International Conference on Machine Learning*.

- Hong, M., Wai, H.-T., Wang, Z., and Yang, Z. (2023). A two-timescale stochastic algorithm framework for bilevel optimization: complexity analysis and application to actor-critic. *SIAM Journal on Optimization*, 33(1):147–180.
- Hu, X., Xiao, N., Liu, X., and Toh, K.-C. (2023). An improved unconstrained approach for bilevel optimization. *SIAM Journal on Optimization*, 33(4):2801–2829.
- Hu, Y., Wang, J., Xie, Y., Krause, A., and Kuhn, D. (2024). Contextual stochastic bilevel optimization. In *Advances in Neural Information Processing Systems*, volume 36.
- Hu, Y., Wang, W., Jia, H., Wang, Y., Chen, Y., Hao, J., Wu, F., and Fan, C. (2020). Learning to utilize shaping rewards: a new approach of reward shaping. In *Advances in Neural Information Processing Systems*, volume 33, pages 15931–15941.
- Huang, F. (2023). On momentum-based gradient methods for bilevel optimization with nonconvex lower-level. *arXiv preprint arXiv:2303.03944*.
- Huang, F., Li, J., Gao, S., and Huang, H. (2022). Enhanced bilevel optimization via Bregman distance. In *Advances in Neural Information Processing Systems*, volume 35, pages 28928–28939.
- Ji, K., Liu, M., Liang, Y., and Ying, L. (2022). Will bilevel optimizers benefit from loops. In *Advances in Neural Information Processing Systems*, volume 35, pages 3011–3023.
- Ji, K., Yang, J., and Liang, Y. (2021). Bilevel optimization: convergence analysis and enhanced design. In *International Conference on Machine Learning*, pages 4882–4892. PMLR.
- Kearns, M. and Singh, S. (1998). Finite-sample convergence rates for Q-learning and indirect algorithms. In *Advances in Neural Information Processing Systems*, volume 11.
- Khanduri, P., Zeng, S., Hong, M., Wai, H.-T., Wang, Z., and Yang, Z. (2021). A near-optimal algorithm for stochastic bilevel optimization via double-momentum. In *Advances in Neural Information Processing Systems*, volume 34, pages 30271–30283.
- Kong, B., Zhu, S., Lu, S., Huang, X., and Yuan, K. (2024). Decentralized bilevel optimization over graphs: loopless algorithmic update and transient iteration complexity. *arXiv preprint arXiv:2402.03167*.
- Kwon, J., Kwon, D., Wright, S., and Nowak, R. D. (2023). A fully first-order method for stochastic bilevel optimization. In *International Conference on Machine Learning*, pages 18083–18113. PMLR.
- Lan, G. (2023). Policy mirror descent for reinforcement learning: linear convergence, new sampling complexity, and generalized problem classes. *Mathematical Programming*, 198(1):1059–1106.
- Lan, Q. (2021). Variational quantum soft actor-critic. *arXiv preprint arXiv:2112.11921*.
- Lan, Q., Tosatto, S., Farrahi, H., and Mahmood, R. (2022). Model-free policy learning with reward gradients. In *International Conference on Artificial Intelligence and Statistics*, pages 4217–4234. PMLR.
- Lee, K., Smith, L., and Abbeel, P. (2021). PEBBLE: feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *International Conference on Machine Learning*.
- Li, G., Wei, Y., Chi, Y., and Chen, Y. (2024a). Breaking the sample size barrier in model-based reinforcement learning with a generative model. *Operations Research*, 72(1):203–221.
- Li, H., Yu, H.-F., Ying, L., and Dhillon, I. S. (2024b). Accelerating primal-dual methods for regularized Markov decision processes. *SIAM Journal on Optimization*, 34(1):764–789.
- Li, J., Gu, B., and Huang, H. (2022). A fully single loop algorithm for bilevel optimization without Hessian inverse. In *Annual AAAI Conference on Artificial Intelligence*, volume 36, pages 7426–7434.
- Li, J., Hu, X., Xu, H., Liu, J., Zhan, X., Jia, Q.-S., and Zhang, Y.-Q. (2023). Mind the gap: offline policy optimization for imperfect rewards. In *International Conference on Learning Representations*.
- Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2016). Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*.
- Liu, H., Simonyan, K., and Yang, Y. (2019). Darts: differentiable architecture search. In *International Conference on Learning Representations*.
- Liu, R., Gao, J., Zhang, J., Meng, D., and Lin, Z. (2021a). Investigating bi-level optimization for learning and vision from a unified perspective: a survey and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):10045–10067.
- Liu, R., Liu, X., Yuan, X., Zeng, S., and Zhang, J. (2021b). A value-function-based interior-point method for non-convex bi-level optimization. In *International Conference on Machine Learning*, pages 6882–6892. PMLR.
- Liu, R., Liu, Y., Yao, W., Zeng, S., and Zhang, J. (2023). Averaged method of multipliers for bi-level optimization without lower-level strong convexity. In *International Conference on Machine Learning*,

- volume 202 of *Proceedings of Machine Learning Research*, pages 21839–21866. PMLR.
- Liu, R., Liu, Y., Zeng, S., and Zhang, J. (2021c). Towards gradient-based bilevel optimization with non-convex followers and beyond. In *Advances in Neural Information Processing Systems*, volume 34, pages 8662–8675.
- Liu, R., Liu, Z., Yao, W., Zeng, S., and Zhang, J. (2024a). Moreau envelope for nonconvex bi-level optimization: a single-loop and Hessian-free solution strategy. In *International Conference on Machine Learning*.
- Liu, R., Mu, P., Yuan, X., Zeng, S., and Zhang, J. (2020). A generic first-order algorithmic framework for bi-level programming beyond lower-level singleton. In *International Conference on Machine Learning*, pages 6305–6315. PMLR.
- Liu, S., Wang, Y., and Gao, X.-S. (2024b). Game-theoretic unlearnable example generator. In *Annual AAAI Conference on Artificial Intelligence*, volume 38, pages 21349–21358.
- Loshchilov, I. and Hutter, F. (2018). Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- Lu, S. (2023). SLM: a smoothed first-order Lagrangian method for structured constrained nonconvex optimization. In *Advances in Neural Information Processing Systems*, volume 36.
- Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., et al. (2021). Isaac Gym: high performance GPU based physics simulation for robot learning. In *Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Mei, J., Xiao, C., Szepesvari, C., and Schuurmans, D. (2020). On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, pages 6820–6829. PMLR.
- Miki, T., Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., and Hutter, M. (2022). Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822.
- Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J. W., Songhori, E., Wang, S., Lee, Y.-J., Johnson, E., Pathak, O., Nazi, A., et al. (2021). A graph placement methodology for fast chip design. *Nature*, 594(7862):207–212.
- Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937. PMLR.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.
- Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. (2017). Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30.
- Pedregosa, F. (2016). Hyperparameter optimization with approximate gradient. In *International Conference on Machine Learning*, pages 737–746. PMLR.
- Saha, A., Pacchiano, A., and Lee, J. (2023). Dueling RL: reinforcement learning with trajectory preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 6263–6289. PMLR.
- Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897. PMLR.
- Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. (2016). High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Seneta, E. (2006). *Non-negative Matrices and Markov Chains*. Springer Science & Business Media.
- Shen, H. and Chen, T. (2023). On penalty-based bilevel gradient descent method. In *International Conference on Machine Learning*. JMLR.
- Shen, H., Yang, Z., and Chen, T. (2024). Principled penalty-based methods for bilevel reinforcement learning and RLHF. In *International Conference on Machine Learning*.
- Shen, H., Zhang, K., Hong, M., and Chen, T. (2023). Towards understanding asynchronous advantage actor-critic: convergence and linear speedup. *IEEE Transactions on Signal Processing*, 71:2579–2594.
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359.
- Song, Z., Lee, J. D., and Yang, Z. (2023). Can we find Nash equilibria at a linear rate in Markov games? In *International Conference on Learning Representations*.
- Sorg, J., Lewis, R. L., and Singh, S. (2010). Reward design via online gradient ascent. In *Advances in Neural Information Processing Systems*, volume 23.

- Sriperumbudur, B. K., Fukumizu, K., Gretton, A., Schölkopf, B., and Lanckriet, G. R. (2009). On integral probability metrics, ϕ -divergences and binary classification. *arXiv preprint arXiv:0901.2698*.
- Sun, J.-M., Yang, J., Mo, K., Lai, Y.-K., Guibas, L., and Gao, L. (2024). Haisor: human-aware indoor scene optimization via deep reinforcement learning. *ACM Transactions on Graphics*, 43(2):1–17.
- Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: an Introduction*. MIT Press.
- Szepesvári, C. (2022). *Algorithms for Reinforcement Learning*. Springer Nature.
- Thoma, V., Pásztor, B., Krause, A., Ramponi, G., and Hu, Y. (2024). Contextual bilevel reinforcement learning for incentive alignment. In *Advances in Neural Information Processing Systems*.
- Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: a physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE.
- Wang, J., Chen, H., Jiang, R., Li, X., and Li, Z. (2021). Fast algorithms for Stackelberg prediction game with least squares loss. In *International Conference on Machine Learning*, pages 10708–10716. PMLR.
- Wang, X., Guo, W., Su, J., Yang, X., and Yan, J. (2022). Zarts: on zero-order optimization for neural architecture search. In *Advances in Neural Information Processing Systems*, volume 35, pages 12868–12880.
- Williams, R. J. and Peng, J. (1991). Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268.
- Wu, J., Chen, S., Wang, M., Wang, H., and Xu, H. (2024). Contractual reinforcement learning: pulling arms with invisible hands. *arXiv preprint arXiv:2407.01458*.
- Yang, H., Luo, L., Li, C. J., Jordan, M., and Fazel, M. (2023). Accelerating inexact hypergradient descent for bilevel optimization. In *OPT 2023: Optimization for Machine Learning*.
- Yang, W., Li, X., and Zhang, Z. (2019). A regularized approach to sparse optimal policy in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 32.
- Yang, Y., Gao, B., and Yuan, Y.-x. (2025). LancBiO: dynamic Lanczos-aided bilevel optimization via Krylov subspace. In *International Conference on Learning Representations*.
- Yao, W., Yu, C., Zeng, S., and Zhang, J. (2024). Constrained bi-level optimization: proximal Lagrangian value function approach and Hessian-free algorithm. In *International Conference on Learning Representations*.
- Ye, J. J., Yuan, X., Zeng, S., and Zhang, J. (2023). Difference of convex algorithms for bilevel programs with applications in hyperparameter selection. *Mathematical Programming*, 198(2):1583–1616.
- Zhan, W., Cen, S., Huang, B., Chen, Y., Lee, J. D., and Chi, Y. (2023). Policy mirror descent for regularized reinforcement learning: a generalized framework with linear convergence. *SIAM Journal on Optimization*, 33(2):1061–1091.
- Zhang, K., Koppel, A., Zhu, H., and Basar, T. (2020). Global convergence of policy gradient methods to (almost) locally optimal policies. *SIAM Journal on Control and Optimization*, 58(6):3586–3612.
- Zheng, Z., Oh, J., and Singh, S. (2018). On learning intrinsic rewards for policy gradient methods. In *Advances in Neural Information Processing Systems*, volume 31.
- Zhu, S., Kong, B., Lu, S., Huang, X., and Yuan, K. (2024). SPARKLE: a unified single-loop primal-dual framework for decentralized bilevel optimization. In *Advances in Neural Information Processing Systems*, volume 37, pages 62912–62987.
- Ziebart, B. D. (2010). *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University.

Checklist

1. For all models and algorithms presented, check if you include:
 - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
 - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] The theoretical results are summarized in Table 1 and Section 7, and detailed analysis is provided in the Appendix.
 - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
2. For any theoretical claim, check if you include:
 - (a) Statements of the full set of assumptions of all theoretical results. [Yes]
 - (b) Complete proofs of all theoretical results. [Yes] Detailed analysis is provided in the Appendix.
 - (c) Clear explanations of any assumptions. [Yes] All the assumptions are clarified in Section 7.

3. For all figures and tables that present empirical results, check if you include:

- (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] We have made the code available on the URL, <https://github.com/UCAS-YanYang/SoBiRL>.
- (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] The experiment details and parameter settings are provided in Appendix J.
- (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
- (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] It is clarified in Appendix J.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

- (a) Citations of the creator If your work uses existing assets. [Yes]
- (b) The license information of the assets, if applicable. [Not Applicable]
- (c) New assets either in the supplemental material or as a URL, if applicable. [Yes] We have made the code available on a URL,
- (d) Information about consent from data providers/curators. [Not Applicable]
- (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

- (a) The full text of instructions given to participants and screenshots. [Not Applicable]
- (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
- (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Bilevel Reinforcement Learning via the Development of Hyper-gradient without Lower-Level Convexity: Supplementary Materials

A RELATED WORK IN BILEVEL OPTIMIZATION

By the implicit function theorem, the approximate implicit differentiation (AID) based method regards the optimal lower-level solution as a function of the upper-level variable, which yields the hyper-gradient to instruct the upper-level update. It implements alternating (inexact) gradient descent steps between two levels (Ghadimi and Wang, 2018; Ji et al., 2021; Chen et al., 2022b; Dagréou et al., 2022; Li et al., 2022; Hong et al., 2023). In addition, generalizing the lower-level strong convexity, the studies (Grazzi et al., 2020, 2021, 2023) focused on the bilevel problem where the lower-level problem is formulated as a fixed-point equation. Moreover, a line of research has been dedicated to the bilevel problem with a nonconvex lower-level objective (Liu et al., 2021c; Shen and Chen, 2023; Huang, 2023; Lu, 2023; Liu et al., 2024a). Specifically, Shen and Chen (2023) proposed a penalty-based method to bypass the requirement of lower-level convexity. In (Huang, 2023), a momentum-based approach was designed to solve the bilevel problem with a lower-level function satisfying the PL condition. Liu et al. (2024a) utilized the Moreau envelope based reformulation to provide a single-loop algorithm for nonconvex-nonconvex bilevel optimization. In stochastic settings, bilevel algorithms have been developed in (Ghadimi and Wang, 2018; Ji et al., 2021; Khanduri et al., 2021; Chen et al., 2022b).

B VECTOR AND MATRIX NOTATION

Throughout this paper, we adhere to the following conventions of vector and matrix notation.

- For any vector $q \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, we denote its component associated with $s \in \mathcal{S}$ as $q_s \in \mathbb{R}^{|\mathcal{A}|}$ or $q(s, \cdot) \in \mathbb{R}^{|\mathcal{A}|}$ in the following way,

$$\forall (s, a) \in \mathcal{S} \times \mathcal{A}, \quad q_s(a) = q(s, a) = q_{sa}.$$

Additionally, for a vector $v \in \mathbb{R}^{|\mathcal{S}|}$, we use the notation of entry $v_s = v(s)$ interchangeably for convenience of exposition.

- $P \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times|\mathcal{S}|}$, $r \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, $\pi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$: the transition matrix, the reward function, and the policy.
- $P^\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$: the transition matrix induced by π .
- $V^\pi \in \mathbb{R}^{|\mathcal{S}|}$, $Q^\pi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$: the soft value functions associated with π .
- $V^*(x) \in \mathbb{R}^{|\mathcal{S}|}$, $Q^*(x) \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, $\pi^*(x) \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$: the optimal soft value functions and the optimal soft policy given $r(x)$.
- $\nabla \phi(x) \in \mathbb{R}^n$: the hyper-gradient.
- $\nabla_x \varphi(x, v) \in \mathbb{R}^{|\mathcal{S}|\times n}$, $\nabla_v \varphi(x, v) \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$: the partial derivatives of $\varphi(x, v)$.
- $\nabla_x f(x, \pi) \in \mathbb{R}^n$, $\nabla_\pi f(x, \pi) \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$: the partial derivatives of $f(x, \pi)$.
- $\nabla r(x) \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times n}$: the derivative of $r(x)$.
- $\nabla V^*(x) \in \mathbb{R}^{|\mathcal{S}|\times n}$, $\nabla Q^*(x) \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times n}$, $\nabla \pi^*(x) \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times n}$: the implicit differentiations.

C PRELIMINARIES ON NUMERICAL LINEAR ALGEBRA

We first revisit some basics for non-negative matrices and Markov chains; see (Seneta, 2006).

Proposition C.1. *Given a matrix $B \in \mathbb{R}^{m \times m}$ satisfying $\|B\|_\infty \leq \gamma$, then the matrix $I - B$ is invertible with the magnitudes of all its eigenvalues located within the interval $[1 - \gamma, 1 + \gamma]$.*

Proof. Denote the entries of B and $I - B$ by $\{b_{ij}\}_{i,j=1}^m$ and $\{l_{ij}\}_{i,j=1}^m$, respectively. Set $\lambda \in \mathbb{C}$ as an eigenvalue of $I - B$, and consider the Gershgorin circle theorem, for $i = 1, 2, \dots, m$,

$$|\lambda - l_{ii}| \leq \sum_{j=1, j \neq i}^m |l_{ij}|. \quad (15)$$

Incorporating $l_{ii} = 1 - b_{ii}$ and $l_{ij} = -b_{ij}$ into (15), we obtain

$$|\lambda - (1 - b_{ii})| \leq \sum_{j=1, j \neq i}^m |b_{ij}|,$$

The absolute value inequality leads to

$$-\sum_{j=1, j \neq i}^m |b_{ij}| \leq |\lambda - 1| - |b_{ii}| \leq \sum_{j=1, j \neq i}^m |b_{ij}|,$$

which completes the proof since

$$\max_{i=1,2,\dots,m} \sum_{j=1}^m |b_{ij}| = \|B\|_\infty = \gamma.$$

□

Definition C.2. A matrix $P \in \mathbb{R}^{m \times m} = (p_{ij})$ is defined to be a transition matrix if all entries are non-negative and each row sums to 1, i.e., for any $i = 1, 2, \dots, m$,

$$\sum_j p_{ij} = 1.$$

Proposition C.3. *A transition matrix $P \in \mathbb{R}^{m \times m} = (p_{ij})$ always has an eigenvalue of 1, while all the remaining eigenvalues have absolute values less than or equal to 1.*

Definition C.4. A matrix $X = (x_{ij}) \in \mathbb{R}^{m \times m}$ is called non-negative if all the entries are equal to or greater than zero, i.e.,

$$x_{ij} \geq 0, \quad \text{for } 1 \leq i, j \leq m.$$

Definition C.5. A matrix $A \in \mathbb{R}^{m \times m}$ is an M -matrix if it can be expressed in the form $A = \mu I - B$, where $B = (b_{ij})$ is non-negative, and $\mu \geq 0$ is equal to or greater than the moduli of any eigenvalue of B .

Proposition C.6. *Given a constant $\gamma \in (0, 1)$ and a transition matrix $P \in \mathbb{R}^{m \times m} = (p_{ij})$, then the matrix $A = I - \gamma P$ is an M -matrix with the moduli of every eigenvalue in $[1 - \gamma, 1 + \gamma]$. Additionally, the inverse of A is*

$$A^{-1} = \sum_{t=0}^{\infty} \gamma^t P^t,$$

which is a non-negative matrix and holds all diagonal entries greater than or equal to 1.

Proof. Combined with Definition C.2 which reveals $\|P\|_\infty = 1$ and Proposition C.1, it follows that any $\lambda \in \mathbb{C}$ as an eigenvalue of A satisfies $|\lambda| \in [1 - \gamma, 1 + \gamma]$, yielding that A is invertible. Since $\rho(\gamma P) = \gamma < 1$ by Proposition C.3, the infinite series $\sum_{t=0}^{\infty} \gamma^t P^t$ converges and coincides with the inverse of A , that is,

$$A^{-1} = \sum_{t=0}^{\infty} \gamma^t P^t. \quad (16)$$

Coupled with P being non-negative and $(\gamma P)^0 = I$, (16) implies that A^{-1} is non-negative with all diagonal entries greater than or equal to 1. □

The next proposition serves as a technical tool in the later analysis.

Proposition C.7. *For any $\pi_1, \pi_2 \in \mathbb{R}_+^{|\mathcal{S}||\mathcal{A}|} \cap \Delta^{\mathcal{S}}$, the vector norms satisfy*

$$\begin{aligned}\|\pi_1 - \pi_2\|_\infty &\leq \|\log \pi_1 - \log \pi_2\|_\infty, \\ \|\pi_1 - \pi_2\|_2 &\leq \|\log \pi_1 - \log \pi_2\|_2.\end{aligned}$$

Proof. It suffices to show for any $0 < a, b < 1$, $|a - b| \leq |\log a - \log b|$. Without loss of generality, suppose that $0 < b \leq a < 1$. Consider the function $G(z) := z - \log z$, with the derivative

$$G'(z) = 1 - \frac{1}{z} < 0, \quad \text{for } z \in (0, 1),$$

which means $G(z)$ is monotonically decreasing on $(0, 1)$. It leads to $a - \log a \leq b - \log b$, equivalently, $a - b \leq \log a - \log b$. \square

D PROOF IN SECTION 4

The structure of $\varphi(\cdot, \cdot)$ facilitates the characterization of $\nabla V^*(x)$, as outlined in the following proposition.

Proposition D.1. *For any $x \in \mathbb{R}^n$, $\varphi(x, \cdot)$ is a contraction map, i.e., $\|\nabla_v \varphi(x, v)\|_\infty = \gamma < 1$, and the matrix $I - \nabla_v \varphi(x, v)$ is invertible. Consequently, $V^*(x)$ is the unique fixed point of $\varphi(x, \cdot)$, with a well-defined derivative $\nabla V^*(x)$, given by*

$$\nabla V^*(x) = (I - \nabla_v \varphi(x, V^*(x)))^{-1} \nabla_x \varphi(x, V^*(x)). \quad (17)$$

Additionally, $\nabla_v \varphi(x, V^*(x))$ coincides with the γ -scaled transition matrix induced by the optimal soft policy $\pi^*(x)$, i.e.,

$$\nabla_v \varphi(x, V^*(x)) = \gamma P^{\pi^*(x)}. \quad (18)$$

Proof. Consider the mapping $F : \mathbb{R}^n \times \mathbb{R}^{|\mathcal{S}|} \rightarrow \mathbb{R}^{|\mathcal{S}|}$ defined by

$$F(x, v) := v - \varphi(x, v).$$

Then the partial derivative of $F(x, v)$ with respect to v reads $I - \nabla_v \varphi(x, v)$. Computing

$$(\nabla_v \varphi(x, v))_{ss'} = \frac{\partial \varphi_s(x, v)}{\partial v_{s'}} = \gamma \frac{\sum_a P_{sas'} \exp(\tau^{-1}(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}))}{\sum_a \exp(\tau^{-1}(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}))} \quad (19)$$

leads to

$$\|\nabla_v \varphi(x, v)\|_\infty = \max_s \sum_{s'} |(\nabla_v \varphi(x, v))_{ss'}| = \max_s \sum_{s'} \frac{\gamma \sum_a P_{ass'} \exp(\tau^{-1}(r_{sa}^x + \gamma \sum_{s''} P_{ass''} v_{s''}))}{\sum_a \exp(\tau^{-1}(r_{sa}^x + \gamma \sum_{s''} P_{ass''} v_{s''}))} = \gamma,$$

which means for any $x \in \mathbb{R}^n$, $\varphi(x, \cdot)$ is a contraction mapping, admitting a unique $V^*(x)$ such that $F(x, V^*(x)) = 0$. Additionally, by applying Proposition C.1, we can derive that $\nabla_v F(x, v)$ is invertible for any $(x, v) \in \mathbb{R}^n \times \mathbb{R}^{|\mathcal{S}|}$. Consequently, the implicit function theorem implies that there exists a differentiable function $V^* : \mathbb{R}^n \rightarrow \mathbb{R}^{|\mathcal{S}|}$ such that, for any $v \in \mathbb{R}^{|\mathcal{S}|}$ and $x \in \mathbb{R}^n$,

$$F(x, V^*(x)) = V^*(x) - \varphi(x, V^*(x)) = \mathbf{0},$$

with the derivative $\nabla V^*(x)$ satisfying

$$\nabla V^*(x) - \nabla_x \varphi(x, V^*(x)) - \nabla_v \varphi(x, V^*(x)) \nabla V^*(x) = \mathbf{0}.$$

Rearranging this yields the result (17). On the other hand, take (19) at $(x, V^*(x))$,

$$\begin{aligned}(\nabla_v \varphi(x, v))_{ss'}|_{v=V^*(x)} &= \gamma \frac{\sum_a P_{sas'} \exp(\tau^{-1}(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} V_{s''}^*(x)))}{\sum_a \exp(\tau^{-1}(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} V_{s''}^*(x)))} \\ &= \gamma \frac{\sum_a P_{sas'} \exp(\tau^{-1} Q_{sa}^*(x))}{\exp(\tau^{-1} V_s^*(x))} \quad (20)\end{aligned}$$

$$\begin{aligned}&= \gamma \sum_a P_{sas'} \exp(\tau^{-1} (Q_{sa}^*(x) - V_s^*(x))) \\ &= \gamma \sum_a P_{sas'} \pi_{sa}^*(x), \quad (21)\end{aligned}$$

where (20) comes from (3) and (6), and (21) follows considering (5). Consequently, revisiting the definition of $P^{\pi^*(x)}$ yields the conclusion $\nabla_v \varphi(x, V^*(x)) = \gamma P^{\pi^*(x)}$. \square

E PROOFS IN SECTION 5

In this section, we provide proofs of the propositions in Section 5. The following proposition indicates that the implicit differentiations $\nabla V^*(x)$ and $\nabla Q^*(x)$ can be estimated by sampling the reward gradient under the policy $\pi^*(x)$.

Proposition E.1. *For any $x \in \mathbb{R}^n$, $s \in \mathcal{S}$, and $a \in \mathcal{A}$,*

$$\begin{aligned}\nabla V_s^*(x) &= \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \nabla r_{s_t, a_t}(x) \mid s_0 = s, \pi^*(x), \mathcal{M}_\tau(x) \right], \\ \nabla Q_{sa}^*(x) &= \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \nabla r_{s_t, a_t}(x) \mid s_0 = s, a_0 = a, \pi^*(x), \mathcal{M}_\tau(x) \right].\end{aligned}$$

Proof. Compute the partial derivative

$$\frac{\partial \varphi_s(x, v)}{\partial x_i} = \frac{\sum_a \partial_{x_i} r_{sa}(x) \exp(\tau^{-1}(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}))}{\sum_a \exp(\tau^{-1}(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}))},$$

and substitute $v = V^*(x)$ into it,

$$\nabla_x \varphi_s(x, V^*(x)) = \sum_a \nabla r_{sa}(x) \pi_{sa}^*(x), \quad (22)$$

where we adopt the similar derivation as (21). Applying Proposition C.6 to (18), we obtain

$$(I - \nabla_v \varphi(x, V^*(x)))^{-1} = \sum_{t=0}^{\infty} (\gamma P^{\pi^*(x)})^t. \quad (23)$$

Employ the subscript $s \in \mathcal{S}$ to denote the corresponding row of a matrix. It follows from (17), (22) and (23) that

$$\begin{aligned}\nabla V_s^*(x) &= (I - \nabla_v \varphi(x, V^*(x)))_s^{-1} \nabla_x \varphi(x, V^*(x)) \\ &= \sum_{t=0}^{\infty} (\gamma P^{\pi^*(x)})_s^t \nabla_x \varphi(x, V^*(x)) \\ &= \sum_{t=0}^{\infty} \gamma^t \sum_{s'} P(s_t = s' | s_0 = s, \pi^*(x)) \sum_a \nabla r_{s'a}(x) \pi_{s'a}^*(x) \\ &= \sum_{t=0}^{\infty} \gamma^t \sum_{s', a} P(s_t = s', a_t = a | s_0 = s, \pi^*(x)) \nabla r_{s'a}(x) \\ &= \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \nabla_x r_{s_t a_t}(x) \mid s_0 = s, \pi^*(x), \mathcal{M}_\tau(x) \right],\end{aligned}$$

where $P(s_t = s', a_t = a | s_0 = s, \pi^*(x))$ is the probability of taking action a at state s' at time t , given that the process starts from state $s_0 = s$ in the MDP $\mathcal{M}_\tau(x)$. Differentiating both sides of

$$Q_{sa}^*(x) = r_{sa}(x) + \gamma \sum_{s'} P_{sas'} V_{s'}^*(x)$$

with respect to x , we have

$$\begin{aligned}\nabla Q_{sa}^*(x) &= \nabla r_{sa}(x) + \gamma \sum_{s'} P_{sas'} \nabla V_{s'}^*(x) \\ &= \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \nabla_x r_{s_t a_t}(x) \mid s_0 = s, a_0 = a, \pi^*(x), \mathcal{M}_\tau(x) \right].\end{aligned}$$

\square

Subsequently, the next proposition characterizes the hyper-gradient via fully first-order information.

Proposition E.2. *The hyper-gradient $\nabla\phi(x)$ of the bilevel reinforcement learning problem (4) with the upper-level function (11) can be computed by*

$$\begin{aligned}\nabla\phi(x) &= \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} [\nabla l(d_1, d_2, \dots, d_I; x)] \\ &\quad + \tau^{-1} \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} \left[l(d_1, d_2, \dots, d_I; x) \left(\sum_i \sum_h \nabla \left(Q_{s_h^i a_h^i}^*(x) - V_{s_h^i}^*(x) \right) \right) \right].\end{aligned}$$

Proof. Recall that

$$P(d_i; \pi^*(x)) = \bar{\rho}(s_0^i) \left[\prod_{h=0}^{H-2} \pi_{s_h^i, a_h^i}^*(x) \bar{P}(s_{h+1}^i | s_h^i, a_h^i) \right] \pi_{s_{H-1}^i, a_{H-1}^i}^*(x),$$

and

$$\phi(x) = \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} [l(d_1, d_2, \dots, d_I; x)] = \sum_{(d_1, \dots, d_I)} l(d_1, d_2, \dots, d_I; x) \Pi_{i=1}^I P(d_i; \pi^*(x)).$$

It follows that

$$\begin{aligned}\nabla\phi(x) &= \sum_{(d_1, \dots, d_I)} \nabla l(d_1, d_2, \dots, d_I; x) \Pi_{i=1}^I P(d_i; \pi^*(x)) + \sum_{(d_1, \dots, d_I)} l(d_1, d_2, \dots, d_I; x) \nabla \Pi_{i=1}^I P(d_i; \pi^*(x)) \\ &= \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} [\nabla l(d_1, d_2, \dots, d_I; x)] + \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} [l(d_1, d_2, \dots, d_I; x) \nabla \log \Pi_{i=1}^I P(d_i; \pi^*(x))] \\ &= \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} [\nabla l(d_1, d_2, \dots, d_I; x)] + \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} \left[l(d_1, d_2, \dots, d_I; x) \nabla \left(\sum_i \log P(d_i; \pi^*(x)) \right) \right] \\ &= \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} [\nabla l(d_1, d_2, \dots, d_I; x)] + \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} \left[l(d_1, d_2, \dots, d_I; x) \left(\sum_i \sum_h \nabla \log \pi_{s_h^i a_h^i}^*(x) \right) \right] \\ &= \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} [\nabla l(d_1, d_2, \dots, d_I; x)] \\ &\quad + \tau^{-1} \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} \left[l(d_1, d_2, \dots, d_I; x) \left(\sum_i \sum_h \nabla \left(Q_{s_h^i a_h^i}^*(x) - V_{s_h^i}^*(x) \right) \right) \right],\end{aligned}$$

where the first equality comes from the chain rules, the second equality is obtained by “log probability tricks” in REINFORCE (Sutton and Barto, 2018), and the last equality takes advantage of (5). \square

F PROPERTIES OF $V^*(x)$ AND $\pi^*(x)$

At the beginning, we uncover the Lipschitz properties of $V^*(x)$, based on the results of Propositions E.1 and H.5. The proof of Proposition H.5 is deferred to Appendix H, since it needs more analytical tools, and this section primarily concentrates on the properties of $V^*(x)$ and $\pi^*(x)$. It is worth stressing that all results in this section only rely on Assumption 7.3.

Lemma F.1. *Under Assumption 7.3, $V^*(x)$ is L_V -Lipschitz continuous with $L_V = \frac{\sqrt{|\mathcal{S}|} C_{rx}}{1-\gamma}$, i.e., for any $x_1, x_2 \in \mathbb{R}^n$,*

$$\|V^*(x_1) - V^*(x_2)\|_2 \leq L_V \|x_1 - x_2\|_2.$$

Proof. Combining the characterization from Proposition E.1

$$\nabla V_s^*(x) = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \nabla r_{s_t, a_t}(x) | s_0 = s, \pi^*(x), \mathcal{M}_\tau(x) \right],$$

and the boundedness of $\nabla r(x)$ revealed by Assumption 7.3 yields

$$\|\nabla V_s^*(x)\|_2 \leq \frac{C_{rx}}{1-\gamma}, \quad \text{for } s \in \mathcal{S}.$$

Put differently, the 2-norm of each row vector in $\nabla V^*(x)$ does not exceed $\frac{C_{rx}}{1-\gamma}$, which means

$$\|\nabla V^*(x)\|_2 \leq \|\nabla V^*(x)\|_F \leq \frac{\sqrt{|\mathcal{S}|} C_{rx}}{1-\gamma}.$$

Here, $\|\cdot\|_F$ is the Frobenius norm of a matrix. Setting $L_V = \frac{\sqrt{|\mathcal{S}|} C_{rx}}{1-\gamma}$ completes the proof. \square

The Lipschitz continuity of $\pi^*(x)$ is connected to that of the value functions by the consistency condition (5).

Proposition F.2. *Under Assumption 7.3, $\pi^*(x)$ is L_π -Lipschitz continuous, i.e., for any $x_1, x_2 \in \mathbb{R}^n$,*

$$\|\pi^*(x_1) - \pi^*(x_2)\|_2 \leq \|\log \pi^*(x_1) - \log \pi^*(x_2)\|_2 \leq L_\pi \|x_1 - x_2\|,$$

where $L_\pi = \frac{2\sqrt{|\mathcal{S}|} |\mathcal{A}| C_{rx}}{\tau(1-\gamma)}$.

Proof. For any $s \in \mathcal{S}$, $a \in \mathcal{A}$, from the results of Proposition E.1,

$$\|\nabla Q_{sa}^*(x)\|_2 \leq \frac{C_{rx}}{1-\gamma} \quad \text{and} \quad \|\nabla V_s^*(x)\|_2 \leq \frac{C_{rx}}{1-\gamma},$$

and the consistency condition (5),

$$\log \pi_{sa}^*(x) = \tau^{-1} (Q_{sa}^*(x) - V_s^*(x)),$$

it follows that

$$\begin{aligned} \|\log \pi_{sa}^*(x_1) - \log \pi_{sa}^*(x_2)\| &\leq \tau^{-1} (\|Q_{sa}^*(x_1) - Q_{sa}^*(x_2)\|_2 + \|V_s^*(x_1) - V_s^*(x_2)\|_2) \\ &\leq \frac{2C_{rx}}{\tau(1-\gamma)} \|x_1 - x_2\|_2. \end{aligned}$$

The shape of $\pi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ implies

$$\|\log \pi^*(x_1) - \log \pi^*(x_2)\|_2 \leq \frac{2\sqrt{|\mathcal{S}|} |\mathcal{A}| C_{rx}}{\tau(1-\gamma)} \|x_1 - x_2\|_2.$$

Applying Proposition C.7 achieves the conclusion. \square

We derive the Lipschitz constant L_{V1} in Proposition H.5 focusing on the smoothness of each $V_s^*(x)$, which lays a foundation for the subsequent proposition. The following Lipschitz constant L_{V1}^M is noted with the superscript M to emphasize we consider the continuity of the whole matrix $\nabla V^*(x)$ here.

Proposition F.3. *Under Assumption 7.3, $V^*(x)$ is L_{V1}^M -Lipschitz smooth, i.e., for any $x_1, x_2 \in \mathbb{R}^n$,*

$$\|\nabla V^*(x_1) - \nabla V^*(x_2)\|_2 \leq L_{V1}^M \|x_1 - x_2\|_2,$$

where $L_{V1}^M = \sqrt{|\mathcal{S}|} \left(\frac{(1+\gamma)C_{rx}L_\pi\sqrt{|\mathcal{A}|}}{(1-\gamma)^2} + \frac{L_r}{1-\gamma} \right)$.

Proof. Proposition H.5 implies that for any $s \in \mathcal{S}$,

$$\|\nabla V_s^*(x_1) - \nabla V_s^*(x_2)\|_2 \leq L_{V1} \|x_1 - x_2\|_2,$$

with $L_{V1} = \frac{(1+\gamma)C_{rx}L_\pi\sqrt{|\mathcal{A}|}}{(1-\gamma)^2} + \frac{L_r}{1-\gamma}$. The property of matrix norms, $\|A\|_2 \leq \|A\|_F$ leads to

$$\|\nabla V^*(x_1) - \nabla V^*(x_2)\|_2 \leq \|\nabla V^*(x_1) - \nabla V^*(x_2)\|_F \leq \sqrt{|\mathcal{S}|} L_{V1} \|x_1 - x_2\|_2.$$

In this way, we can choose

$$L_{V1}^M = \sqrt{|\mathcal{S}|} L_{V1} = \sqrt{|\mathcal{S}|} \left(\frac{(1+\gamma)C_{rx}L_\pi\sqrt{|\mathcal{A}|}}{(1-\gamma)^2} + \frac{L_r}{1-\gamma} \right).$$

\square

Algorithm 2 Model-based soft bilevel reinforcement learning algorithm (M-SoBiRL)

Input: outer and inner iteration numbers K, N , step sizes β, η , initialization x_0, π_0, Q_0, w_0

```

1: for  $k = 1, \dots, K$  do
2:    $A_k = (I - \gamma P^{\pi_k})^\top$ ,  $b_k = U^\top \text{diag}(\pi_k) \nabla_\pi f(x_k, \pi_k)$ 
3:   Compute  $V_k$  by (9)
4:    $w_k = w_{k-1} - \eta ((A_k^\top A_k) w_{k-1} - A_k^\top b_k)$ 
5:    $x_{k+1} = x_k - \beta \hat{\nabla} \phi(x_k, \pi_k, V_k, w_k)$ 
6:    $Q_{k,0} = Q_k$ 
7:   for  $n = 1, \dots, N$  do
8:      $Q_{k,n} = \mathcal{T}_{\mathcal{M}_\tau(x_{k+1})}(Q_{k,n-1})$ 
9:   end for
10:   $Q_{k+1} = Q_{k,N}$ ,  $\pi_{k+1} = \text{softmax}(Q_{k+1}/\tau)$ 
11: end for
Output:  $(x_{K+1}, \pi_K)$ 

```

G ANALYSIS OF M-SOBIRL

Enlightened by the hyper-gradient,

$$\begin{aligned}
 \nabla \phi(x) &= \nabla_x f(x, \pi^*(x)) + \nabla \pi^*(x)^\top \nabla_\pi f(x, \pi^*(x)) \\
 &= \nabla_x f(x, \pi^*(x)) + \tau^{-1} \nabla r(x)^\top \text{diag}(\pi^*(x)) \nabla_\pi f(x, \pi^*(x)) \\
 &\quad - \tau^{-1} \nabla_x \varphi(x, V^*(x))^\top (I - \gamma P^{\pi^*(x)})^{-\top} U^\top \text{diag}(\pi^*(x)) \nabla_\pi f(x, \pi^*(x)),
 \end{aligned} \tag{24}$$

we propose the model-based algorithm, M-SoBiRL; see Algorithm 2. Note that in practice, the special structure of RL will facilitate the computation. Specifically, the matrix $A_k = (I - \gamma P^{\pi_k})^\top$ is usually sparse in RL, and the calculation of $b_k = U^\top \text{diag}(\pi_k) \nabla_\pi f(x_k, \pi_k)$ also involves a sparse constant matrix U and a diagonal $\text{diag}(\pi_k)$. Therefore, the multiplications $A_k^\top A_k$ and $A_k^\top b_k$ for updating w_k in line 4 can be accelerated by sparse multiplication oracles, the cost of which is denoted as c_{sparse} . Additionally, evaluating $\hat{\nabla} \phi$ and updating x in line 5 cost $\mathcal{O}(|\mathcal{S}||\mathcal{A}|n)$. Moreover, the other lines in Algorithm 2 totally cost $\mathcal{O}(|\mathcal{S}||\mathcal{A}|)$ since they are element-wise operations on Q_k or V_k . In summary, the computation cost of each outer iteration is $\mathcal{O}(|\mathcal{S}||\mathcal{A}|n + c_{\text{sparse}})$.

Next, we focus on the convergence analysis for M-SoBiRL, starting with a short proof sketch for guidance.

G.1 Proof Sketch of Theorem 7.4

Denote $\pi_k^* := \pi^*(x_k)$, $Q_k^* := Q^*(x_k)$, $V_k^* := V^*(x_k)$, and

$$w_k^* := (I - \gamma P^{\pi_k^*})^{-\top} U^\top \text{diag}(\pi_k^*) \nabla_\pi f(x_k, \pi_k^*).$$

The proof is structured into four main steps.

Step1: Preliminary Properties

Appendix G.2 studies the Lipschitz and boundedness properties of the quantities related to Algorithm 2, laying a foundation for the following analysis.

Step2: Upper-bounding the Residual $\|w_k - w_k^*\|_2$

Appendix G.3 bounds the term $\|w_k - w_k^*\|_2$,

$$\begin{aligned}
 \|w_k - w_k^*\|_2^2 &\leq \left(1 - \frac{\eta(1-\gamma)^2}{2}\right) \|w_{k-1} - w_{k-1}^*\|_2^2 + \eta^2 C_{\sigma\pi}^2 \left(\frac{1}{(1-\gamma)^2} + 4\right) \|\pi_k - \pi_k^*\|_\infty^2 \\
 &\quad + 2L_w^2 \left(1 + \frac{1}{\eta(1-\gamma)^2}\right) \|x_k - x_{k-1}\|_2^2 + 4\eta^2 |\mathcal{S}|^2 (1+\gamma)^2 \|w_{k-1} - w_{k-1}^*\|_2^2,
 \end{aligned}$$

where the coefficient $\left(1 - \frac{\eta(1-\gamma)^2}{2}\right)$ implies its descent property, and the constants $C_{\sigma\pi}, L_w$ will be specified later.

Step3: Upper-bounding the Errors $\|Q_k - Q_k^*\|_\infty$

Appendix G.4 measures the quality of the policy evaluation, $\|Q_k - Q_k^*\|_\infty$, which can be bounded by

$$\|Q_k - Q_k^*\|_\infty \leq \gamma^N \left(\|Q_{k-1} - Q_{k-1}^*\|_\infty + \frac{C_{rx}}{1-\gamma} \|x_k - x_{k-1}\|_2 \right),$$

with the contraction factor γ^N responsible for the convergence property.

Step4: Assembling the Estimations above and Achieving the Conclusion

Considering the merit function

$$\mathcal{L}_k := \phi(x_k) + \zeta_w \|w_k - w_k^*\|_2^2 + \zeta_Q \|Q_k - Q_k^*\|_2^2,$$

where the coefficients ζ_w and ζ_Q are to be determined later, Appendix G.5 evaluates $\mathcal{L}_{k+1} - \mathcal{L}_k$ to reveal the decreasing property of \mathcal{L}_k . As a result, it proves that M-SoBiRL enjoys the convergence rate $\mathcal{O}(\epsilon^{-1})$ with the inner iteration number $N = \mathcal{O}(1)$ independent of the solution accuracy ϵ .

G.2 Lipschitz Properties of Quantities Related to Algorithm 2

In this subsection, we study the Lipschitz and boundedness properties of the quantities related to Algorithm 2, setting the groundwork for convergence analysis.

Lemma G.1. P^π is $\sqrt{|\mathcal{A}|}$ -Lipschitz continuous with respect to π , i.e., for any π^1, π^2 ,

$$\|P^{\pi^1} - P^{\pi^2}\|_2 \leq \sqrt{|\mathcal{A}|} \|\pi^1 - \pi^2\|_2$$

Proof. The element of $P^{\pi^1} - P^{\pi^2}$ in position (s, s') is $\sum_a (\pi_{sa}^1 - \pi_{sa}^2) P_{sas'}$. Consider the 2-norm of the s -th row in $P^{\pi^1} - P^{\pi^2}$,

$$\begin{aligned} & \sum_{s'} \left(\sum_a (\pi_{sa}^1 - \pi_{sa}^2) P_{sas'} \right)^2 \\ & \leq \sum_{s'} \left(\sum_a (\pi_{sa}^1 - \pi_{sa}^2)^2 \right) \left(\sum_a P_{sas'}^2 \right) \\ & \leq \left(\sum_a (\pi_{sa}^1 - \pi_{sa}^2)^2 \right) \sum_{s'} \left(\sum_a P_{sas'} \right) \\ & = |\mathcal{A}| \|\pi_s^1 - \pi_s^2\|_2^2, \end{aligned}$$

which leads to

$$\|P^{\pi^1} - P^{\pi^2}\|_2 \leq \|P^{\pi^1} - P^{\pi^2}\|_F \leq \sqrt{|\mathcal{A}|} \|\pi^1 - \pi^2\|_2.$$

□

Lemma G.2. $U \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|}$ has full column rank, and

$$\|U\|_2 \in [\sqrt{|\mathcal{A}|}(1+\gamma), \sqrt{|\mathcal{S}||\mathcal{A}|(1+\gamma)}].$$

Proof. Define that for any $a \in \mathcal{A}$, $U_a = I - \gamma P_a$, where $P_a \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ with $P_a(s, s') = P_{sas'}$. By the triangle inequality, $\|U_a\|_\infty \in [1 - \gamma, 1 + \gamma]$. Subsequently, the property of the matrix norms, $\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty}$, produces

$$\|U_a\|_2 \leq \sqrt{\|U_a\|_1 \|U_a\|_\infty} \leq \sqrt{|\mathcal{S}|(1+\gamma)}. \quad (25)$$

Applying Proposition C.1 to U_a deduces that the magnitude of every eigenvalue of U_a belongs to $[1 - \gamma, 1 + \gamma]$. The property that $\|U_a\|_2$ is greater than the magnitude of any of its eigenvalues leads to

$$\|U_a\|_2 \geq 1 + \gamma. \quad (26)$$

The definition of U_a reveals that the $|\mathcal{S}|$ rows of the matrix U_a are distributed in U with a spacing of $|\mathcal{A}|$. In this way, U has full column rank since U_a is invertible. Consider the singular value of $U \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|}$. Specifically,

$$U^\top U = \left(\sum_{a \in \mathcal{A}} \left(\sum_{s' \in \mathcal{S}} U_a^\top(s') U_a(s') \right) \right) = \left(\sum_{a \in \mathcal{A}} U_a^\top U_a \right),$$

where $U_a(s')$ denotes the s' -th row of U_a . Combining (25) with (26), it follows that

$$\|U\|_2 \in \left[\sqrt{|\mathcal{A}|}(1 + \gamma), \sqrt{|\mathcal{S}||\mathcal{A}|(1 + \gamma)} \right].$$

□

Lemma G.3. Define

$$\vartheta(x) := U^\top \text{diag}(\pi^*(x)) \nabla_\pi f(x, \pi^*(x)).$$

Under Assumptions 7.1 and 7.3, $\vartheta(x)$ is L_ϑ -Lipschitz continuous, i.e., for any $x_1, x_2 \in \mathbb{R}^n$,

$$\|\vartheta(x_1) - \vartheta(x_2)\|_2 \leq L_\vartheta \|x_1 - x_2\|_2,$$

where $L_\vartheta = \sqrt{|\mathcal{S}||\mathcal{A}|(1 + \gamma)}(L_f(1 + L_\pi) + C_{f\pi}L_\pi)$.

Proof. Given the boundedness of $\nabla_\pi f(x, \pi^*(x))$ from Assumption 7.1 and the boundedness of $\|U\|_2$ derived in Lemma G.2, it follows that

$$\begin{aligned} & \|\vartheta(x_1) - \vartheta(x_2)\|_2 \\ & \leq \sqrt{|\mathcal{S}||\mathcal{A}|(1 + \gamma)} \|\text{diag}(\pi_1^*) \nabla_\pi f(x_1, \pi_1^*) - \text{diag}(\pi_2^*) \nabla_\pi f(x_2, \pi_2^*)\|_2 \\ & \leq \sqrt{|\mathcal{S}||\mathcal{A}|(1 + \gamma)} \|\text{diag}(\pi_1^*)(\nabla_\pi f(x_1, \pi_1^*) - \nabla_\pi f(x_2, \pi_2^*))\|_2 \\ & \quad + \sqrt{|\mathcal{S}||\mathcal{A}|(1 + \gamma)} \|(\text{diag}(\pi_1^*) - \text{diag}(\pi_2^*)) \nabla_\pi f(x_2, \pi_2^*)\|_2 \\ & \leq \sqrt{|\mathcal{S}||\mathcal{A}|(1 + \gamma)} (L_f (\|x_1 - x_2\|_2 + \|\pi_1^* - \pi_2^*\|_\infty) + C_{f\pi} \|\pi_1^* - \pi_2^*\|_\infty) \\ & \leq \sqrt{|\mathcal{S}||\mathcal{A}|(1 + \gamma)} (L_f (1 + L_\pi) + C_{f\pi}L_\pi) \|x_1 - x_2\|_2. \end{aligned}$$

□

Consequently, we concentrate on the Lipschitz property of the hyper-objective $\phi(x)$, which plays a vital role in analyzing the convergence properties of Algorithm 2. The superscript M is appended to the Lipschitz constant, L_ϕ^M to make a distinction from the constant used in the model-free scenario.

Proposition G.4. Under Assumptions 7.1 and 7.3, $\nabla\phi(x)$ is L_ϕ^M -Lipschitz continuous, i.e., for any $x_1, x_2 \in \mathbb{R}^n$,

$$\|\nabla\phi(x_1) - \nabla\phi(x_2)\|_2 \leq L_\phi^M \|x_1 - x_2\|_2,$$

where $L_\phi^M = \mathcal{O}\left(\frac{|\mathcal{S}|^{3/2}|\mathcal{A}|^{3/2}}{(1-\gamma)^3}\right)$ is specified in (31).

Proof. Revisit the expression of $\nabla\phi(x)$ and incorporate the notation $\vartheta(x)$,

$$\begin{aligned} \nabla\phi(x) &= \nabla_x f(x, \pi^*(x)) + \tau^{-1} \left(\nabla r(x) - U \nabla V^*(x) \right)^\top \text{diag}(\pi^*(x)) \nabla_\pi f(x, \pi^*(x)) \\ &= \nabla_x f(x, \pi^*(x)) + \tau^{-1} \nabla r(x)^\top \text{diag}(\pi^*(x)) \nabla_\pi f(x, \pi^*(x)) - \tau^{-1} \nabla V^*(x)^\top \vartheta(x). \end{aligned} \tag{27}$$

Consider the first term in (27),

$$\begin{aligned} & \|\nabla_x f(x_1, \pi^*(x_1)) - \nabla_x f(x_2, \pi^*(x_2))\|_2 \\ & \leq L_f (\|x_1 - x_2\|_2 + \|\pi^*(x_1) - \pi^*(x_2)\|_2) \\ & \leq (L_f + L_f L_\pi) \|x_1 - x_2\|_2. \end{aligned} \tag{28}$$

Subsequent analysis involves the Lipschitz property of the second term,

$$\begin{aligned}
 & \left\| \nabla r(x_1)^\top \text{diag}(\pi^*(x_1)) \nabla_\pi f(x_1, \pi^*(x_1)) - \nabla r(x_2)^\top \text{diag}(\pi^*(x_2)) \nabla_\pi f(x_2, \pi^*(x_2)) \right\|_2 \\
 & \leq \left\| \nabla r(x_1)^\top \text{diag}(\pi^*(x_1)) \nabla_\pi f(x_1, \pi^*(x_1)) - \nabla r(x_2)^\top \text{diag}(\pi^*(x_1)) \nabla_\pi f(x_1, \pi^*(x_1)) \right\|_2 \\
 & \quad + \left\| \nabla r(x_2)^\top \text{diag}(\pi^*(x_1)) \nabla_\pi f(x_1, \pi^*(x_1)) - \nabla r(x_2)^\top \text{diag}(\pi^*(x_2)) \nabla_\pi f(x_1, \pi^*(x_1)) \right\|_2 \\
 & \quad + \left\| \nabla r(x_2)^\top \text{diag}(\pi^*(x_2)) \nabla_\pi f(x_1, \pi^*(x_1)) - \nabla r(x_2)^\top \text{diag}(\pi^*(x_2)) \nabla_\pi f(x_2, \pi^*(x_2)) \right\|_2 \\
 & \leq \sqrt{|\mathcal{S}| |\mathcal{A}|} L_r C_{f\pi} \|x_1 - x_2\|_2 + \sqrt{|\mathcal{S}| |\mathcal{A}|} C_{rx} C_{f\pi} L_\pi \|x_1 - x_2\|_2 \\
 & \quad + \sqrt{|\mathcal{S}| |\mathcal{A}|} C_{rx} (L_f + L_f L_\pi) \|x_1 - x_2\|_2 \\
 & = \sqrt{|\mathcal{S}| |\mathcal{A}|} (L_r C_{f\pi} + C_{rx} C_{f\pi} L_\pi + C_{rx} (L_f + L_f L_\pi)) \|x_1 - x_2\|_2. \tag{29}
 \end{aligned}$$

Similarly, the third term in (27) satisfy

$$\begin{aligned}
 & \left\| \nabla V^*(x_1)^\top \vartheta(x_1) - \nabla V^*(x_2)^\top \vartheta(x_2) \right\|_2 \\
 & \leq \left\| \nabla V^*(x_1)^\top \vartheta(x_1) - \nabla V^*(x_1)^\top \vartheta(x_2) \right\|_2 + \left\| \nabla V^*(x_1)^\top \vartheta(x_2) - \nabla V^*(x_2)^\top \vartheta(x_2) \right\|_2 \\
 & \leq \frac{\sqrt{|\mathcal{S}|} C_{rx}}{1 - \gamma} L_\vartheta \|x_1 - x_2\|_2 + \sqrt{|\mathcal{S}| |\mathcal{A}| (1 + \gamma)} C_{f\pi} L_{V1}^M \|x_1 - x_2\|_2 \\
 & = \left(\frac{\sqrt{|\mathcal{S}|} C_{rx}}{1 - \gamma} L_\vartheta + \sqrt{|\mathcal{S}| |\mathcal{A}| (1 + \gamma)} C_{f\pi} L_{V1}^M \right) \|x_1 - x_2\|_2. \tag{30}
 \end{aligned}$$

Collecting (28), (29) and (30) concludes that $\nabla \phi(x)$ is L_ϕ^M -Lipschitz continuous with

$$\begin{aligned}
 L_\phi^M &= L_f + L_f L_\pi + \sqrt{|\mathcal{S}| |\mathcal{A}|} (L_r C_{f\pi} + C_{rx} C_{f\pi} L_\pi + C_{rx} (L_f + L_f L_\pi)) \\
 &\quad + \frac{\sqrt{|\mathcal{S}|} C_{rx}}{1 - \gamma} L_\vartheta + \sqrt{|\mathcal{S}| |\mathcal{A}| (1 + \gamma)} C_{f\pi} L_{V1}^M \\
 &= \mathcal{O}\left(\frac{|\mathcal{S}|^{\frac{3}{2}} |\mathcal{A}|^{\frac{3}{2}}}{(1 - \gamma)^3}\right), \tag{31}
 \end{aligned}$$

where (31) comes from the expressions of L_π , L_ϑ and L_{V1}^M . \square

G.3 Convergence Property of w_k

Define

$$A_k^* := \left(I - \gamma P^{\pi^*(x_k)} \right)^\top, \quad \text{and} \quad b_k^* := \vartheta(x_k).$$

In Algorithm 2, the goal of w_k is to track $w_k^* = (A_k^*)^{-1} b_k^*$ through outer iterations. This section delves into the descent property of $\|w_k - w_k^*\|_2$. Initially, we prove that the exact quantities $\|b_k^*\|_2$ and $\|w_k^*\|_2$ are uniformly bounded.

Lemma G.5. *Under Assumptions 7.1 and 7.3, it holds that*

$$\begin{aligned}
 \|b_k^*\|_2 &\leq \sqrt{|\mathcal{S}| |\mathcal{A}| (1 + \gamma)} C_{f\pi}, \\
 \|w_k^*\|_2 &\leq \frac{\sqrt{|\mathcal{S}| |\mathcal{A}| (1 + \gamma)}}{1 - \gamma} C_{f\pi}.
 \end{aligned}$$

Proof. Taking into account $\|U\|_2 \leq \sqrt{|\mathcal{S}| |\mathcal{A}| (1 + \gamma)}$, $\pi^* \in \Delta^{|\mathcal{S}|}$, and $\nabla_\pi f(x, \pi^*(x)) \leq C_{f\pi}$, we can obtain

$$\begin{aligned}
 \|b_k^*\|_2 &= \|U^\top \text{diag}(\pi_k^*) \nabla_\pi f(x_k, \pi_k^*)\| \\
 &\leq \|U\|_2 \|\text{diag}(\pi_k^*)\|_2 \|\nabla_\pi f(x_k, \pi_k^*)\|_2 \\
 &\leq \sqrt{|\mathcal{S}| |\mathcal{A}| (1 + \gamma)} C_{f\pi},
 \end{aligned}$$

where $\pi_k^* = \pi^*(x_k)$. Additionally, denote the eigenvalue of A_k^* with the smallest moduli by $\lambda_{\min}(A_k^*)$. Proposition C.6 reveals $\lambda_{\min}(A_k^*) \geq 1 - \gamma$. In this way, $\|A_k^*\|_2 \geq \lambda_{\min}(A_k^*)$ and

$$\|(A_k^*)^{-1}\|_2 \leq \frac{1}{\|A_k^*\|_2} \leq \frac{1}{1 - \gamma}.$$

So, it completes the proof by

$$\|w_k^*\|_2 = \|(A_k^*)^{-1} b_k^*\|_2 \leq \|(A_k^*)^{-1}\|_2 \|b_k^*\|_2 \leq \frac{\sqrt{|\mathcal{S}| |\mathcal{A}| (1 + \gamma)}}{1 - \gamma} C_{f\pi}.$$

□

Lemma G.6. Under Assumptions 7.1 and 7.3, the following inequalities hold in Algorithm 2.

$$\begin{aligned} \|A_k^* - A_{k-1}^*\|_2 &\leq \sqrt{|\mathcal{A}|} L_\pi \|x_k - x_{k-1}\|_2, \\ \|A_k - A_k^*\|_2 &\leq \sqrt{|\mathcal{A}|} \|\pi_k - \pi_k^*\|_2, \\ \|A_k\|_2 &\leq \sqrt{|\mathcal{S}| (1 + \gamma)}, \\ \|A_k^*\|_2 &\leq \sqrt{|\mathcal{S}| (1 + \gamma)}, \\ \left\| A_k^\top A_k - (A_k^*)^\top A_k^* \right\|_2 &\leq 2\sqrt{|\mathcal{S}| |\mathcal{A}| (1 + \gamma)} \|\pi_k - \pi_k^*\|_2, \\ \|b_k - b_k^*\|_2 &\leq \sqrt{|\mathcal{S}| |\mathcal{A}| (1 + \gamma)} (L_f + C_{f\pi}) \|\pi_k - \pi_k^*\|_\infty. \end{aligned}$$

Proof. Drawing on Lemma G.1 and the boundedness of A_k, A_k^* ,

$$\|A_k\|_2 \leq \sqrt{\|A_k\|_1 \|A_k\|_\infty} \leq \sqrt{|\mathcal{S}| (1 + \gamma)}, \quad \text{and} \quad \|A_k^*\|_2 \leq \sqrt{|\mathcal{S}| (1 + \gamma)},$$

we obtain

$$\begin{aligned} \|A_k - A_k^*\|_2 &\leq \sqrt{|\mathcal{A}|} \|\pi_k - \pi_k^*\|_2, \\ \|A_k^* - A_{k-1}^*\|_2 &\leq \left\| P^{\pi_k^*} - P^{\pi_{k-1}^*} \right\|_2 \leq \sqrt{|\mathcal{A}|} \|\pi_k^* - \pi_{k-1}^*\|_2 \leq \sqrt{|\mathcal{A}|} L_\pi \|x_k - x_{k-1}\|_2, \end{aligned}$$

and

$$\begin{aligned} \left\| A_k^\top A_k - (A_k^*)^\top A_k^* \right\|_2 &\leq \|A_k^\top (A_k - A_k^*)\|_2 + \left\| (A_k - A_k^*)^\top A_k^* \right\|_2 \\ &\leq 2\sqrt{|\mathcal{S}| |\mathcal{A}| (1 + \gamma)} \|\pi_k - \pi_k^*\|_2. \end{aligned}$$

Moreover, the Lipschitz and boundedness properties related to ∇f in Assumption 7.1 reveal that

$$\begin{aligned} \|b_k - b_k^*\|_2 &\leq \sqrt{|\mathcal{S}| |\mathcal{A}| (1 + \gamma)} \|\text{diag}(\pi_k) \nabla_\pi f(x_k, \pi_k) - \text{diag}(\pi_k^*) \nabla_\pi f(x_k, \pi_k^*)\|_2 \\ &\leq \sqrt{|\mathcal{S}| |\mathcal{A}| (1 + \gamma)} \|\text{diag}(\pi_k) (\nabla_\pi f(x_k, \pi_k) - \nabla_\pi f(x_k, \pi_k^*))\|_2 \\ &\quad + \sqrt{|\mathcal{S}| |\mathcal{A}| (1 + \gamma)} \|(\text{diag}(\pi_k) - \text{diag}(\pi_k^*)) \nabla_\pi f(x_k, \pi_k^*)\|_2 \\ &\leq \sqrt{|\mathcal{S}| |\mathcal{A}| (1 + \gamma)} (L_f + C_{f\pi}) \|\pi_k - \pi_k^*\|_\infty. \end{aligned}$$

□

The following lemma illustrates the descent property of $\|w_k - w_k^*\|_2^2$

Lemma G.7. Under Assumption 7.1 and 7.3, the iterates $\{(x_k, \pi_k, w_k)\}$ generated by Algorithm 2 satisfy

$$\begin{aligned} \|w_k - w_k^*\|_2^2 &\leq \left(1 - \frac{\eta(1 - \gamma)^2}{2}\right) \|w_{k-1} - w_{k-1}^*\|_2^2 + \eta^2 C_{\sigma\pi}^2 \left(\frac{1}{(1 - \gamma)^2} + 4\right) \|\pi_k - \pi_k^*\|_\infty^2 \\ &\quad + 2L_w^2 \left(1 + \frac{1}{\eta(1 - \gamma)^2}\right) \|x_k - x_{k-1}\|_2^2 + 4\eta^2 |\mathcal{S}|^2 (1 + \gamma)^2 \|w_{k-1} - w_{k-1}^*\|_2^2, \end{aligned}$$

with

$$C_{\sigma\pi} = \frac{2|\mathcal{S}|^{\frac{3}{2}}|\mathcal{A}|^{\frac{3}{2}}(1+\gamma)C_{f\pi}}{1-\gamma} + |\mathcal{S}||\mathcal{A}|^{\frac{3}{2}}\sqrt{(1+\gamma)}C_{f\pi} + |\mathcal{S}|\sqrt{|\mathcal{A}|}(1+\gamma)(L_f + C_{f\pi}),$$

$$L_w = \frac{1}{1-\gamma} \left(L_\vartheta + \frac{|\mathcal{A}|}{1-\gamma}\sqrt{|\mathcal{S}|(1+\gamma)}C_{f\pi}L_\pi \right).$$

Proof. By the identity $\|a - b\|_2^2 = \|a\|^2 + \|b\|^2 - 2\langle a, b \rangle$, we establish

$$\|w_k - w_k^*\|_2^2 = \|w_k - w_{k-1}^*\|_2^2 + \|w_{k-1}^* - w_k^*\|_2^2 - 2\langle w_k - w_{k-1}^*, w_k^* - w_{k-1}^* \rangle, \quad (32)$$

and estimate each member. Denote

$$\sigma_k := A_k^\top A_k w_{k-1} - A_k^\top b_k,$$

which is the update direction of w_{k-1} , and correspondingly, use the following notation for reference,

$$\sigma_k^* := (A_k^*)^\top A_k^* w_k^* - (A_k^*)^\top b_k^*.$$

By the update rule of w , $w_k = w_{k-1} - \eta\sigma_k$, we decompose the first term in (32):

$$\|w_k - w_{k-1}^*\|_2^2 = \|w_{k-1} - w_{k-1}^*\|_2^2 + \eta^2 \|\sigma_k\|_2^2 - 2\eta \langle \sigma_k, w_{k-1} - w_{k-1}^* \rangle. \quad (33)$$

The identity

$$\begin{aligned} \sigma_k - \sigma_k^* &= A_k^\top A_k (w_{k-1} - w_{k-1}^*) + \left(A_k^\top A_k - (A_k^*)^\top A_k^* \right) w_{k-1}^* \\ &\quad + (A_k^* - A_k)^\top b_k^* + A_k(b_k^* - b_k) \end{aligned}$$

and the observation that $\sigma_k^* = 0$ yield

$$\begin{aligned} \|\sigma_k\|_2 &= \|\sigma_k - \sigma_k^*\|_2 \\ &\leq \|A_k^\top A_k (w_{k-1} - w_{k-1}^*)\|_2 + \left\| \left(A_k^\top A_k - (A_k^*)^\top A_k^* \right) w_{k-1}^* \right\|_2 \\ &\quad + \|(A_k^* - A_k)^\top b_k^*\|_2 + \|A_k(b_k^* - b_k)\|_2 \\ &\leq |\mathcal{S}|(1+\gamma)\|w_{k-1} - w_{k-1}^*\|_2 + \frac{2|\mathcal{S}||\mathcal{A}|(1+\gamma)C_{f\pi}}{1-\gamma}\|\pi_k - \pi_k^*\|_2 \\ &\quad + |\mathcal{A}|\sqrt{|\mathcal{S}|(1+\gamma)}C_{f\pi}\|\pi_k - \pi_k^*\|_2 + |\mathcal{S}|\sqrt{|\mathcal{A}|}(1+\gamma)(L_f + C_{f\pi})\|\pi_k - \pi_k^*\|_\infty, \\ &\leq |\mathcal{S}|(1+\gamma)\|w_{k-1} - w_{k-1}^*\|_2 + C_{\sigma\pi}\|\pi_k - \pi_k^*\|_\infty, \end{aligned} \quad (34)$$

with $C_{\sigma\pi} = \frac{2|\mathcal{S}|^{\frac{3}{2}}|\mathcal{A}|^{\frac{3}{2}}(1+\gamma)C_{f\pi}}{1-\gamma} + |\mathcal{S}||\mathcal{A}|^{\frac{3}{2}}\sqrt{(1+\gamma)}C_{f\pi} + |\mathcal{S}|\sqrt{|\mathcal{A}|}(1+\gamma)(L_f + C_{f\pi})$, where the second inequality comes from the results of Lemma G.6, and the last inequality is obtained by $\|\pi_k - \pi_k^*\|_2 \leq \sqrt{|\mathcal{S}||\mathcal{A}|}\|\pi_k - \pi_k^*\|_\infty$. Subsequently, we bound the third term in (32) in a similar way,

$$\begin{aligned} &\langle \sigma_k, w_{k-1} - w_{k-1}^* \rangle \\ &= \langle \sigma_k - \sigma_k^*, w_{k-1} - w_{k-1}^* \rangle \\ &= \langle A_k^\top A_k (w_{k-1} - w_{k-1}^*), w_{k-1} - w_{k-1}^* \rangle + \langle \left(A_k^\top A_k - (A_k^*)^\top A_k^* \right) w_{k-1}^*, w_{k-1} - w_{k-1}^* \rangle \\ &\quad + \langle (A_k^* - A_k)^\top b_k^*, w_{k-1} - w_{k-1}^* \rangle + \langle A_k(b_k^* - b_k), w_{k-1} - w_{k-1}^* \rangle \\ &\geq (1-\gamma)^2 \|w_{k-1} - w_{k-1}^*\|_2^2 - C_{\sigma\pi}\|\pi_k - \pi_k^*\|_\infty \|w_{k-1} - w_{k-1}^*\|_2. \end{aligned}$$

By Young's inequality,

$$\begin{aligned} \eta C_{\sigma\pi} \|\pi_k - \pi_k^*\|_\infty \|w_{k-1} - w_{k-1}^*\|_2 &\leq \frac{\eta^2}{2\rho_1} \|\pi_k - \pi_k^*\|_\infty^2 + \frac{C_{\sigma\pi}^2\rho_1}{2} \|w_{k-1} - w_{k-1}^*\|_2^2 \\ &= \frac{\eta^2 C_{\sigma\pi}^2}{2(1-\gamma)^2} \|\pi_k - \pi_k^*\|_\infty^2 + \frac{(1-\gamma)^2}{2} \|w_{k-1} - w_{k-1}^*\|_2^2, \end{aligned}$$

where we choose $\rho_1 = \frac{(1-\gamma)^2}{C_{\sigma\pi}^2}$. It follows that

$$\eta \langle \sigma_k, w_{k-1} - w_{k-1}^* \rangle \geq \frac{(1-\gamma)^2}{2} \|w_{k-1} - w_{k-1}^*\|_2^2 - \frac{\eta^2 C_{\sigma\pi}^2}{2(1-\gamma)^2} \|\pi_k - \pi_k^*\|_\infty^2. \quad (35)$$

Substituting the inequality (35) into (32) yields

$$\|w_k - w_{k-1}^*\|_2^2 \leq \left(1 - \eta(1-\gamma)^2\right) \|w_{k-1} - w_{k-1}^*\|_2^2 + \eta^2 \|\sigma_k\|_2^2 + \frac{\eta^2 C_{\sigma\pi}^2}{(1-\gamma)^2} \|\pi_k - \pi_k^*\|_\infty^2. \quad (36)$$

Then, we bound the term $\|w_k^* - w_{k-1}^*\|_2$,

$$\begin{aligned} & \|w_k^* - w_{k-1}^*\|_2 \\ &= \left\| (A_k^*)^{-1} b_k^* - (A_{k-1}^*)^{-1} b_{k-1}^* \right\|_2 \\ &\leq \left\| (A_k^*)^{-1} (b_k^* - b_{k-1}^*) \right\|_2 + \left\| (A_k^*)^{-1} (A_k^* - A_{k-1}^*) (A_{k-1}^*)^{-1} b_{k-1}^* \right\|_2 \\ &\leq \frac{L_\vartheta}{1-\gamma} \|x_k - x_{k-1}\|_2 + \frac{|\mathcal{A}|}{(1-\gamma)^2} \sqrt{|\mathcal{S}|(1+\gamma)} C_{f\pi} L_\pi \|x_k - x_{k-1}\|_2 \\ &= L_w \|x_k - x_{k-1}\|_2, \end{aligned} \quad (37)$$

with $L_w = \frac{1}{1-\gamma} \left(L_\vartheta + \frac{|\mathcal{A}|}{1-\gamma} \sqrt{|\mathcal{S}|(1+\gamma)} C_{f\pi} L_\pi \right)$. Apply Young's inequality to the inner product term in (32),

$$\begin{aligned} & -2 \langle w_k - w_{k-1}^*, w_k^* - w_{k-1}^* \rangle \\ &= -2 \langle w_{k-1} - w_{k-1}^*, w_k^* - w_{k-1}^* \rangle + 2\eta \langle \sigma_k, w_k^* - w_{k-1}^* \rangle \\ &\leq \rho_2 \|w_{k-1} - w_{k-1}^*\|_2^2 + \frac{1}{\rho_2} \|w_k^* - w_{k-1}^*\|_2^2 + \eta^2 \|\sigma_k\|_2^2 + \|w_k^* - w_{k-1}^*\|_2^2 \\ &= \frac{\eta(1-\gamma)^2}{2} \|w_{k-1} - w_{k-1}^*\|_2^2 + \frac{2}{\eta(1-\gamma)^2} \|w_k^* - w_{k-1}^*\|_2^2 \\ &\quad + \eta^2 \|\sigma_k\|_2^2 + \|w_k^* - w_{k-1}^*\|_2^2, \end{aligned} \quad (38)$$

where we take $\rho_2 = \frac{\eta(1-\gamma)^2}{2}$. Assembling (36), (37) and (38), we can estimate $\|w_k - w_k^*\|_2^2$ in (32),

$$\begin{aligned} \|w_k - w_k^*\|_2^2 &= \|w_k - w_{k-1}^*\|_2^2 + \|w_{k-1}^* - w_k^*\|_2^2 - 2 \langle w_k - w_{k-1}^*, w_k^* - w_{k-1}^* \rangle \\ &\leq \left(1 - \eta(1-\gamma)^2\right) \|w_{k-1} - w_{k-1}^*\|_2^2 + \eta^2 \|\sigma_k\|_2^2 + \frac{\eta^2 C_{\sigma\pi}^2}{(1-\gamma)^2} \|\pi_k - \pi_k^*\|_\infty^2 \\ &\quad + \|w_{k-1}^* - w_k^*\|_2^2 + \frac{\eta(1-\gamma)^2}{2} \|w_{k-1} - w_{k-1}^*\|_2^2 \\ &\quad + \frac{2}{\eta(1-\gamma)^2} \|w_k^* - w_{k-1}^*\|_2^2 + \eta^2 \|\sigma_k\|_2^2 + \|w_k^* - w_{k-1}^*\|_2^2 \\ &= \left(1 - \frac{\eta(1-\gamma)^2}{2}\right) \|w_{k-1} - w_{k-1}^*\|_2^2 + \frac{\eta^2 C_{\sigma\pi}^2}{(1-\gamma)^2} \|\pi_k - \pi_k^*\|_\infty^2 \\ &\quad + 2 \left(1 + \frac{1}{\eta(1-\gamma)^2}\right) \|w_k^* - w_{k-1}^*\|_2^2 + 2\eta^2 \|\sigma_k\|_2^2 \\ &\leq \left(1 - \frac{\eta(1-\gamma)^2}{2}\right) \|w_{k-1} - w_{k-1}^*\|_2^2 + \eta^2 C_{\sigma\pi}^2 \left(\frac{1}{(1-\gamma)^2} + 4\right) \|\pi_k - \pi_k^*\|_\infty^2 \\ &\quad + 2L_w^2 \left(1 + \frac{1}{\eta(1-\gamma)^2}\right) \|x_k - x_{k-1}\|_2^2 + 4\eta^2 |\mathcal{S}|^2 (1+\gamma)^2 \|w_{k-1} - w_{k-1}^*\|_2^2, \end{aligned}$$

where the last inequality is earned by incorporating

$$\|\sigma_k\|_2^2 \leq 2|\mathcal{S}|^2(1+\gamma)^2\|w_{k-1} - w_{k-1}^*\|_2^2 + 2C_{\sigma\pi}^2\|\pi_k - \pi_k^*\|_\infty^2$$

and

$$\|w_k^* - w_{k-1}^*\|_2^2 \leq L_w^2\|x_k - x_{k-1}\|_2^2,$$

implied in (34) and (37), respectively. \square

G.4 Convergence Properties of π_k and Q_k

The softmax mapping, plays a significant role in the update rule of π_k and Q_k , for which we begin this section by introducing some properties of it, following from (Cen et al., 2022). Given a $\theta \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, for any $s \in \mathcal{S}$, use $\theta_s \in \mathbb{R}^{|\mathcal{A}|}$ to denote a vector with $\theta_s(a) = \theta_{sa} = \theta(s, a)$. Recall the softmax mapping:

$$\begin{aligned} \text{softmax} : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} &\longrightarrow \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \\ \theta &\longmapsto \left(\frac{\exp(\theta(s, a))}{\sum_{a'} \exp(\theta(s, a'))} \right)_{sa}. \end{aligned}$$

Consider the typical component, where $\pi_q \in \mathbb{R}^{|\mathcal{A}|}$ is parameterized by $q \in \mathbb{R}^{|\mathcal{A}|}$,

$$\pi_q(a) = \frac{\exp(q(a))}{\sum_{a'} \exp(q(a'))}.$$

In this way,

$$\begin{aligned} & |\log(\|\exp(q_1)\|_1) - \log(\|\exp(q_2)\|_1)| \\ &= \left| \langle q_1 - q_2, \nabla_{\theta} \log \|\exp(q)\|_1 \rangle_{q=q_c} \right| \\ &\leq \|q_1 - q_2\|_\infty \left\| \nabla_q \log \|\exp(q)\|_1 \right\|_{q=q_c} \\ &= \|q_1 - q_2\|_\infty, \end{aligned} \tag{39}$$

where q_c is a certain convex combination of q_1 and q_2 , and

$$\nabla_q \log \|\exp(q)\|_1 = \frac{1}{\|\exp(q)\|_1} \exp(q) = \pi_q.$$

It follows that

$$\begin{aligned} \|\log \pi_{q_1} - \log \pi_{q_2}\|_\infty &\leq \|q_1 - q_2\|_\infty + |\log(\|\exp(q_1)\|_1) - \log(\|\exp(q_2)\|_1)| \\ &\leq 2\|q_1 - q_2\|_\infty. \end{aligned} \tag{40}$$

In Algorithm 2, it adopts Q_k to approximate the optimal soft Q-value function of π_k dynamically, i.e., the environment $\mathcal{M}_\tau(x_k)$ based on which the value function is evaluated varies through the outer iterations. To this end, we will analyze the difference of value functions, incurred by the change of the environment parameterized by x . Recall that $\pi_k^* = \pi^*(x_k)$, $Q_k^* = Q^*(x_k)$, $V_k^* = V^*(x_k)$.

Lemma G.8. *Under Assumption 7.2, given a policy π , for any $x_1, x_2 \in \mathbb{R}^n$,*

$$\|V^\pi(x_1) - V^\pi(x_2)\|_\infty \leq \|Q^\pi(x_1) - Q^\pi(x_2)\|_\infty \leq \frac{C_{rx}}{1-\gamma} \|x_1 - x_2\|_2,$$

Proof. By definitions of the soft value functions, for any $s \in \mathcal{S}$ and $a \in \mathcal{A}$,

$$\begin{aligned} |V_s^\pi(x_1) - V_s^\pi(x_2)| &= \mathbb{E}_{a \sim \pi(\cdot|s)} [(-\tau \log \pi(a|s) + Q_{sa}^\pi(x_1)) - (-\tau \log \pi(a|s) + Q_{sa}^\pi(x_2))] \\ &\leq \|Q^\pi(x_1) - Q^\pi(x_2)\|_\infty, \end{aligned} \tag{41}$$

and

$$|Q_{sa}^\pi(x_1) - Q_{sa}^\pi(x_2)| \leq |r_{sa}(x_1) - r_{sa}(x_2)| + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} [|V_{s'}^\pi(x_1) - V_{s'}^\pi(x_2)|]. \quad (42)$$

Substituting (41) into (42) yields

$$\|Q^\pi(x_1) - Q^\pi(x_2)\|_\infty \leq C_{rx} \|x_1 - x_2\|_2 + \gamma \|Q^\pi(x_1) - Q^\pi(x_2)\|_\infty,$$

which leads to the conclusion. \square

We present the properties of the operator $\mathcal{T}_{\mathcal{M}_\tau(x)} : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \mapsto \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, following from (Haarnoja et al., 2018a; Cen et al., 2022).

Proposition G.9. *The soft Bellman optimality operator $\mathcal{T}_{\mathcal{M}_\tau(x)} : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \mapsto \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ associated with $x \in \mathbb{R}^n$ satisfies the properties below.*

- The optimal soft Q -function $Q^*(x)$ is a fixed point of $\mathcal{T}_{\mathcal{M}_\tau(x)}$, i.e.,

$$\mathcal{T}_{\mathcal{M}_\tau(x)}(Q^*(x)) = Q^*(x). \quad (43)$$

- $\mathcal{T}_{\mathcal{M}_\tau(x)}$ is a γ -contraction in the ℓ_∞ norm, i.e., for any $Q_1, Q_2 \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, it holds that

$$\|\mathcal{T}_{\mathcal{M}_\tau(x)}(Q^1) - \mathcal{T}_{\mathcal{M}_\tau(x)}(Q^2)\|_\infty \leq \gamma \|Q^1 - Q^2\|_\infty. \quad (44)$$

- With any initial Q_0 , applying $\mathcal{T}_{\mathcal{M}_\tau(x)}$ repeatedly converges to $Q^*(x)$ linearly, i.e., for any $N \in \mathbb{N}$,

$$\left\| \mathcal{T}_{\mathcal{M}_\tau(x)}^N(Q^0) - Q^*(x) \right\|_\infty \leq \gamma^N \|Q^0 - Q^*(x)\|_\infty.$$

Proof. Substitute one of the consistency conditions from (Nachum et al., 2017),

$$V_s^*(x) = \tau \log (\|\exp(Q_s^*(x)/\tau)\|_1),$$

into the definition of $\mathcal{T}_{\mathcal{M}_\tau(x)}$,

$$\begin{aligned} \mathcal{T}_{\mathcal{M}_\tau(x)}(Q^*(x))(s, a) &= r_{sa}(x) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} [\tau \log (\|\exp(Q_s^*(x)/\tau)\|_1)] \\ &= r_{sa}(x) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} [V_s^*(x)] \\ &= Q^*(x)(s, a). \end{aligned}$$

Additionally, for any $Q^1, Q^2 \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$,

$$\begin{aligned} &|\mathcal{T}_{\mathcal{M}_\tau(x)}(Q^1)(s, a) - \mathcal{T}_{\mathcal{M}_\tau(x)}(Q^2)(s, a)| \\ &= \gamma \tau \left| \mathbb{E}_{s' \sim P(\cdot|s,a)} [\log (\|\exp(Q_s^1(x)/\tau)\|_1)] - \mathbb{E}_{s' \sim P(\cdot|s,a)} [\log (\|\exp(Q_s^2(x)/\tau)\|_1)] \right| \\ &\leq \gamma \tau \|Q^1/\tau - Q^2/\tau\|_\infty \\ &= \gamma \|Q^1 - Q^2\|_\infty, \end{aligned}$$

where the inequality follows from (39). Combine (43) and (44),

$$\begin{aligned} \left\| \mathcal{T}_{\mathcal{M}_\tau(x)}^N(Q^0) - Q^*(x) \right\|_\infty &= \left\| \mathcal{T}_{\mathcal{M}_\tau(x)}^N(Q^0) - \mathcal{T}_{\mathcal{M}_\tau(x)}^N(Q^*(x)) \right\|_\infty \\ &\leq \gamma \left\| \mathcal{T}_{\mathcal{M}_\tau(x)}^{N-1}(Q^0) - \mathcal{T}_{\mathcal{M}_\tau(x)}^{N-1}(Q^*(x)) \right\|_\infty \\ &\leq \gamma^2 \left\| \mathcal{T}_{\mathcal{M}_\tau(x)}^{N-2}(Q^0) - \mathcal{T}_{\mathcal{M}_\tau(x)}^{N-2}(Q^*(x)) \right\|_\infty \\ &\leq \dots \\ &\leq \gamma^N \|Q^0 - Q^*(x)\|_\infty. \end{aligned}$$

\square

Combining the contraction property of $\mathcal{T}_{\mathcal{M}_\tau(x)}$ and Lemma G.8 yield

$$\|Q_{k+1} - Q_{k+1}^*\|_\infty \leq \gamma^N \|Q_k - Q_{k+1}^*\|_\infty \leq \gamma^N \left(\|Q_k - Q_k^*\|_\infty + \frac{C_{rx}}{1-\gamma} \|x_{k+1} - x_k\|_2 \right). \quad (45)$$

Revisiting the update rule for V_k ,

$$\begin{aligned} \forall s \in \mathcal{S} : V_k(s) &= \tau \log \left(\sum_a \exp(Q_k(s, a) / \tau) \right) \\ &= \tau \log (\|Q_k(s)/\tau\|_1), \end{aligned}$$

we apply the inequality (39) and the consistency condition (7),

$$\begin{aligned} |V_k(s) - V_s^*(x_k)| &= \tau |\log(\|Q_k(s)/\tau\|_1) - \log(\|Q_k^*(s)/\tau\|_1)| \\ &\leq \tau \|Q_k(s)/\tau - Q_k^*(s)/\tau\|_\infty \\ &= \|Q_k(s) - Q_k^*(s)\|_\infty. \end{aligned}$$

In this way, we bound the error term,

$$\|V_k - V_k^*\|_\infty \leq \|Q_k - Q_k^*\|_\infty. \quad (46)$$

G.5 Convergence Analysis of M-SoBiRL

To begin with, some lemmas are provided for measuring the quality of the hyper-gradient estimator $\widehat{\nabla}\phi$. Following this, we proceed to the convergence analysis of Algorithm 2.

Lemma G.10. *Under Assumptions 7.1 and 7.3, the following inequalities hold in Algorithm 2.*

$$\begin{aligned} \|\text{diag}(\pi_k) \nabla_\pi f(x_k, \pi_k) - \text{diag}(\pi_k^*) \nabla_\pi f(x_k, \pi_k^*)\|_2 &\leq (L_f + C_{f\pi}) \|\pi_k - \pi_k^*\|_\infty \\ \|\nabla_x \varphi(x_k, V_k)^\top w_k - \nabla_x \varphi(x_k, V_k^*)^\top w_k^*\|_2 &\leq \sqrt{|\mathcal{S}|} C_{rx} \|w_k - w_k^*\|_2 \\ &\quad + \frac{2|\mathcal{S}||\mathcal{A}|^{3/2} \sqrt{(1+\gamma)} C_{f\pi} C_{rx}}{\tau(1-\gamma)} \|V_k^* - V_k\|_\infty. \end{aligned}$$

Proof. Drawing on the Lipschitz and boundedness properties of f in Assumption 7.1, one can establish

$$\begin{aligned} &\|\text{diag}(\pi_k) \nabla_\pi f(x_k, \pi_k) - \text{diag}(\pi_k^*) \nabla_\pi f(x_k, \pi_k^*)\|_2 \\ &\leq \|\text{diag}(\pi_k) (\nabla_\pi f(x_k, \pi_k) - \nabla_\pi f(x_k, \pi_k^*))\|_2 + \|(\text{diag}(\pi_k) - \text{diag}(\pi_k^*)) \nabla_\pi f(x_k, \pi_k^*)\|_2 \\ &\leq L_f \|\pi_k - \pi_k^*\|_\infty + C_{f\pi} \|\pi_k - \pi_k^*\|_\infty \\ &= (L_f + C_{f\pi}) \|\pi_k - \pi_k^*\|_\infty. \end{aligned}$$

Compute the partial derivative

$$\frac{\partial \varphi_s(x, v)}{\partial x_i} = \frac{\sum_a \partial_{x_i} r_{sa}(x) \exp(\tau^{-1}(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}))}{\sum_a \exp(\tau^{-1}(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}))},$$

and substitute $v = V^*(x)$ into it,

$$\nabla_x \varphi_s(x, V^*(x)) = \sum_a \nabla r_{sa}(x) \pi_{sa}^*(x).$$

Denote the auxiliary policy generated by softmax parameterization associated with V , i.e., for any $(s, a) \in \mathcal{S} \times \mathcal{A}$,

$$\pi_{sa}^V = \frac{\exp(\tau^{-1}(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} V_{s''}))}{\sum_a \exp(\tau^{-1}(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} V_{s''}))}.$$

In a similar fashion,

$$\nabla_x \varphi_s(x, V) = \sum_a \nabla r_{sa}(x) \pi_{sa}^V(x).$$

Applying Lemma H.2 with $H = 1, I = 1$, we obtain

$$\|\nabla_x \varphi_s(x, V^*(x)) - \nabla_x \varphi_s(x, V)\|_2 \leq C_{rx} \sqrt{|\mathcal{A}|} \|\pi_s^*(x) - \pi_s^V\|_2.$$

Collecting all rows of $\nabla_x \varphi(x, v)$ leads to

$$\|\nabla_x \varphi(x, V)\|_2 \leq \|\nabla_x \varphi(x, V)\|_F \leq \sqrt{|\mathcal{S}|} C_{rx},$$

and

$$\begin{aligned} & \|\nabla_x \varphi(x, V^*(x)) - \nabla_x \varphi(x, V)\|_2 \\ & \leq \|\nabla_x \varphi(x, V^*(x)) - \nabla_x \varphi(x, V)\|_F \\ & \leq C_{rx} \sqrt{|\mathcal{A}|} \|\pi^*(x) - \pi^V\|_2 \\ & \leq \sqrt{|\mathcal{S}|} |\mathcal{A}| C_{rx} \|\pi^*(x) - \pi^V\|_\infty \\ & \leq 2\sqrt{|\mathcal{S}|} |\mathcal{A}| \tau^{-1} C_{rx} \|V^*(x) - V\|_\infty, \end{aligned}$$

where the last inequality results from (40). It follows that

$$\begin{aligned} & \|\nabla_x \varphi(x_k, V_k)^\top w_k - \nabla_x \varphi(x_k, V_k^*)^\top w_k^*\|_2 \\ & \leq \|\nabla_x \varphi(x_k, V_k)^\top (w_k - w_k^*)\|_2 + \|(\nabla_x \varphi(x_k, V_k) - \nabla_x \varphi(x_k, V_k^*))^\top w_k^*\|_2 \\ & \leq \|\nabla_x \varphi(x_k, V_k)\|_2 \|w_k - w_k^*\|_2 + \|\nabla_x \varphi(x_k, V_k) - \nabla_x \varphi(x_k, V_k^*)\|_2 \|w_k^*\|_2 \\ & \leq \sqrt{|\mathcal{S}|} C_{rx} \|w_k - w_k^*\|_2 + \frac{2|\mathcal{S}| |\mathcal{A}|^{3/2} \sqrt{(1+\gamma)} C_{f\pi} C_{rx}}{\tau(1-\gamma)} \|V_k^* - V_k\|_\infty. \end{aligned}$$

□

In the sequel, we analyze the error introduced by the estimator $\hat{\nabla}\phi(x, \pi, V, w)$ in approximating the true hyper-gradient $\nabla\phi(x)$, as stated in the following lemma.

Lemma G.11. *Under Assumptions 7.1 and 7.3, the hyper-gradient estimator constructed in Algorithm 2,*

$$\hat{\nabla}\phi(x_k, \pi_k, V_k, w_k) = \nabla_x f(x_k, \pi_k) + \frac{1}{\tau} \nabla r(x_k)^\top \text{diag}(\pi_k) \nabla_\pi f(x_k, \pi_k) - \frac{1}{\tau} \nabla_x \varphi(x_k, V_k)^\top w_k,$$

satisfy

$$\begin{aligned} & \|\hat{\nabla}\phi(x_k, \pi_k, V_k, w_k) - \nabla\phi(x_k)\|_2 \\ & \leq L_{\hat{\phi}\pi} \|\pi_k - \pi_k^*\|_\infty + \sqrt{|\mathcal{S}|} C_{rx} \|w_k - w_k^*\|_2 + \frac{2|\mathcal{S}| |\mathcal{A}|^{3/2} \sqrt{(1+\gamma)} C_{f\pi} C_{rx}}{\tau(1-\gamma)} \|V_k^* - V_k\|_\infty. \end{aligned}$$

with

$$L_{\hat{\phi}\pi} = L_f + \tau^{-1} \sqrt{|\mathcal{S}| |\mathcal{A}|} C_{rx} (L_f + C_{f\pi}).$$

Proof. Recall the counterpart true hyper-gradient,

$$\nabla\phi(x_k) = \nabla_x f(x_k, \pi_k^*) + \tau^{-1} (\nabla r_k)^\top \text{diag}(\pi_k^*) \nabla_\pi f(x_k, \pi_k^*) - \tau^{-1} \nabla_x \varphi(x_k, V_k^*)^\top w_k^*.$$

Take into account the difference and substitute the results of Lemma G.10,

$$\begin{aligned}
 & \left\| \widehat{\nabla} \phi_k(x_k, \pi_k, V_k, w_k) - \nabla \phi(x_k) \right\|_2 \\
 & \leq \| \nabla_x f(x_k, \pi_k^*) - \nabla_x f(x_k, \pi_k) \|_2 + \tau^{-1} \| \nabla r_k \|_2 \| \text{diag}(\pi_k) \nabla_\pi f(x_k, \pi_k) - \text{diag}(\pi_k^*) \nabla_\pi f(x_k, \pi_k^*) \|_2 \\
 & \quad + \tau^{-1} \| \nabla_x \varphi(x_k, v_k)^\top w_k - \nabla_x \varphi(x_k, v_k^*)^\top w_k^* \|_2 \\
 & \leq L_f \| \pi_k - \pi_k^* \|_\infty + \tau^{-1} \sqrt{|\mathcal{S}| |\mathcal{A}|} C_{rx} (L_f + C_{f\pi}) \| \pi_k - \pi_k^* \|_\infty \\
 & \quad + \sqrt{|\mathcal{S}|} C_{rx} \| w_k - w_k^* \|_2 + \frac{2 |\mathcal{S}| |\mathcal{A}|^{3/2} \sqrt{(1+\gamma)} C_{f\pi} C_{rx}}{\tau (1-\gamma)} \| V_k^* - V_k \|_\infty \\
 & = \left(L_f + \tau^{-1} \sqrt{|\mathcal{S}| |\mathcal{A}|} C_{rx} (L_f + C_{f\pi}) \right) \| \pi_k - \pi_k^* \|_\infty \\
 & \quad + \sqrt{|\mathcal{S}|} C_{rx} \| w_k - w_k^* \|_2 + \frac{2 |\mathcal{S}| |\mathcal{A}|^{3/2} \sqrt{(1+\gamma)} C_{f\pi} C_{rx}}{\tau (1-\gamma)} \| V_k^* - V_k \|_\infty.
 \end{aligned}$$

□

Consequently, we arrive at the convergence analysis. Denote $\delta_Q^k := \| Q_k - Q_k^* \|_\infty^2$, $\delta_\pi^k := \| \log \pi_k - \log \pi_k^* \|_\infty^2$ and $\delta_w^k := \| w_k - w_k^* \|_2^2$. In this fashion, revisit the established results in previous subsections.

- The definition of `softmax` and the estimation (40) reveal that

$$\delta_\pi^k \leq \frac{4}{\tau^2} \delta_Q^k. \quad (47)$$

- Apply (47) on the result of Lemma G.7,

$$\begin{aligned}
 \delta_w^{k+1} & \leq \left(1 - \frac{\eta(1-\gamma)^2}{2} + 4\eta^2 |\mathcal{S}|^2 (1+\gamma)^2 \right) \delta_w^k + \frac{4\eta^2 C_{\sigma\pi}^2}{\tau^2} \left(\frac{1}{(1-\gamma)^2} + 4 \right) \delta_Q^{k+1} \\
 & \quad + 2L_w^2 \left(1 + \frac{1}{\eta(1-\gamma)^2} \right) \beta^2 \left\| \widehat{\nabla} \phi_k \right\|_2^2
 \end{aligned} \quad (48)$$

- Substitute the update rule of x into (45),

$$\delta_Q^{k+1} \leq 2\gamma^{2N} \left(\delta_Q^k + \left(\frac{C_{rx}}{1-\gamma} \right)^2 \beta^2 \left\| \widehat{\nabla} \phi_k \right\|_2^2 \right). \quad (49)$$

- Incorporate (46),

$$\| V_k - V_k^* \|_\infty \leq \| Q_k - Q_k^* \|_\infty$$

into the estimation of Lemma G.11,

$$\begin{aligned}
 \left\| \widehat{\nabla} \phi_k - \nabla \phi(x_k) \right\|_2^2 & \leq 3L_{\phi\pi}^2 \delta_\pi^k + 3 |\mathcal{S}| C_{rx}^2 \delta_w^k + \frac{12 |\mathcal{S}|^2 |\mathcal{A}|^3 (1+\gamma) C_{f\pi}^2 C_{rx}^2}{\tau^2 (1-\gamma)^2} \delta_Q^k \\
 & \leq \frac{12}{\tau^2} \left(L_{\phi\pi}^2 + \frac{|\mathcal{S}|^2 |\mathcal{A}|^3 (1+\gamma) C_{f\pi}^2 C_{rx}^2}{(1-\gamma)^2} \right) \delta_Q^k + 3 |\mathcal{S}| C_{rx}^2 \delta_w^k.
 \end{aligned} \quad (50)$$

Theorem G.12. Under Assumptions 7.1 and 7.3, in Algorithm 2, we can choose constant step sizes β, η , and the inner iteration number $N \sim \mathcal{O}(1)$. Then the iterates $\{x_k\}$ satisfy

$$\frac{1}{K} \sum_{k=1}^K \| \nabla \phi(x_k) \|_2^2 = \mathcal{O} \left(\frac{1}{K} \right).$$

Detailed parameter setting is listed as (56).

Proof. We consider the merit function

$$\mathcal{L}_k := \phi(x_k) + \zeta_Q \delta_Q^k + \zeta_w \delta_w^k,$$

where the coefficients ζ_Q and ζ_w are to be determined later. By the Lipschitz property of $\phi(x)$ and the update rule of x in Algorithm 2,

$$\begin{aligned} \phi(x_{k+1}) &\leq \phi(x_k) - \beta \left\langle \widehat{\nabla} \phi_k, \nabla \phi(x_k) \right\rangle + \frac{L_\phi^M}{2} \beta^2 \left\| \widehat{\nabla} \phi_k \right\|_2^2 \\ &= \phi(x_k) - \frac{\beta}{2} \left(\left\| \widehat{\nabla} \phi_k \right\|_2^2 + \left\| \nabla \phi(x_k) \right\|_2^2 - \left\| \widehat{\nabla} \phi_k - \nabla \phi(x_k) \right\|_2^2 \right) + \frac{L_\phi^M}{2} \beta^2 \left\| \widehat{\nabla} \phi_k \right\|_2^2 \\ &\leq \phi(x_k) - \frac{\beta}{2} \left\| \nabla \phi(x_k) \right\|_2^2 + \frac{\beta}{2} \left\| \widehat{\nabla} \phi_k - \nabla \phi(x_k) \right\|_2^2 + \left(\frac{L_\phi^M}{2} \beta^2 - \frac{\beta}{2} \right) \left\| \widehat{\nabla} \phi_k \right\|_2^2. \end{aligned} \quad (51)$$

Subsequently, it follows that

$$\begin{aligned} \mathcal{L}_{k+1} - \mathcal{L}_k &= \phi(x_{k+1}) + \zeta_Q \delta_Q^{k+1} + \zeta_w \delta_w^{k+1} - \phi(x_k) - \zeta_Q \delta_Q^k - \zeta_w \delta_w^k \\ &\leq -\frac{\beta}{2} \left\| \nabla \phi(x_k) \right\|_2^2 + \frac{\beta}{2} \left\| \widehat{\nabla} \phi_k - \nabla \phi(x_k) \right\|_2^2 + \left(\frac{L_\phi^M}{2} \beta^2 - \frac{\beta}{2} \right) \left\| \widehat{\nabla} \phi_k \right\|_2^2 \\ &\quad + \zeta_Q \delta_Q^{k+1} + \zeta_w \delta_w^{k+1} - \zeta_Q \delta_Q^k - \zeta_w \delta_w^k \end{aligned} \quad (52)$$

$$\begin{aligned} &\leq -\frac{\beta}{2} \left\| \nabla \phi(x_k) \right\|_2^2 + \frac{\beta}{2} \left(\frac{12}{\tau^2} \left(L_{\phi\pi}^2 + \frac{|\mathcal{S}|^2 |\mathcal{A}|^3 (1+\gamma) C_{f\pi}^2 C_{rx}^2}{(1-\gamma)^2} \right) \delta_Q^k + 3 |\mathcal{S}| C_{rx}^2 \delta_w^k \right) \\ &\quad + \left(\frac{L_\phi^M}{2} \beta^2 - \frac{\beta}{2} \right) \left\| \widehat{\nabla} \phi_k \right\|_2^2 \end{aligned}$$

$$\begin{aligned} &\quad + 2\gamma^{2N} \left(\zeta_Q + \frac{4\eta^2 C_{\sigma\pi}^2}{\tau^2} \zeta_w \left(\frac{1}{(1-\gamma)^2} + 4 \right) \right) \left(\delta_Q^k + \left(\frac{C_{rx}}{1-\gamma} \right)^2 \beta^2 \left\| \widehat{\nabla} \phi_k \right\|_2^2 \right) \\ &\quad + \zeta_w \left(1 - \frac{\eta(1-\gamma)^2}{2} + 4\eta^2 |\mathcal{S}|^2 (1+\gamma)^2 \right) \delta_w^k \\ &\quad + 2L_w^2 \zeta_w \left(1 + \frac{1}{\eta(1-\gamma)^2} \right) \beta^2 \left\| \widehat{\nabla} \phi_k \right\|_2^2 - \zeta_Q \delta_Q^k - \zeta_w \delta_w^k \end{aligned} \quad (53)$$

$$= -\frac{\beta}{2} \left\| \nabla \phi(x_k) \right\|_2^2 + I_Q \delta_Q^k + I_w \delta_w^k + I_{\widehat{\phi}} \left\| \widehat{\nabla} \phi_k \right\|_2^2, \quad (54)$$

with

$$\begin{aligned} I_w &= \frac{3\beta |\mathcal{S}| C_{rx}^2}{2} + \zeta_w \left(-\frac{\eta(1-\gamma)^2}{2} + 4\eta^2 |\mathcal{S}|^2 (1+\gamma)^2 \right) \\ I_Q &= 2\gamma^{2N} \left(\zeta_Q + \frac{4\eta^2 C_{\sigma\pi}^2}{\tau^2} \zeta_w \left(\frac{1}{(1-\gamma)^2} + 4 \right) \right) - \zeta_Q \\ &\quad + \frac{6\beta}{\tau^2} \left(L_{\phi\pi}^2 + \frac{|\mathcal{S}|^2 |\mathcal{A}|^3 (1+\gamma) C_{f\pi}^2 C_{rx}^2}{(1-\gamma)^2} \right) \\ I_{\widehat{\phi}} &= \frac{L_\phi^M}{2} \beta^2 + 2\beta^2 L_w^2 \zeta_w \left(1 + \frac{1}{\eta(1-\gamma)^2} \right) - \frac{\beta}{2} \\ &\quad + 2\gamma^{2N} \left(\zeta_Q + \frac{4\eta^2 C_{\sigma\pi}^2}{\tau^2} \zeta_w \left(\frac{1}{(1-\gamma)^2} + 4 \right) \right) \left(\frac{C_{rx}}{1-\gamma} \right)^2 \beta^2, \end{aligned}$$

where (52) comes from (51) and (53) is the derivation of substituting (49), (48) and (50). One sufficient condition for $I_Q, I_w, I_{\hat{\phi}} < 0$ is that

$$\begin{cases} \zeta_Q = \zeta_w = 1, \\ \frac{3\beta|\mathcal{S}|C_{rx}^2}{2\eta(1-\gamma)^2} < \frac{1}{2} - 4\eta|\mathcal{S}|^2 \frac{(1+\gamma)^2}{(1-\gamma)^2}, \\ \beta < \frac{\tau^2}{8} \left(L_{\hat{\phi}\pi}^2 + \frac{|\mathcal{S}|^2|\mathcal{A}|^3(1+\gamma)C_{f\pi}^2C_{rx}^2}{(1-\gamma)^2} \right)^{-1}, \\ \gamma^{2N} < \frac{1}{8} \left(1 + \frac{4\eta^2C_{\sigma\pi}^2}{\tau^2} \left(\frac{1}{(1-\gamma)^2} + 4 \right) \right)^{-1}, \\ \frac{L_{\phi}^M}{2}\beta + 2\beta L_w^2 \left(1 + \frac{1}{\eta(1-\gamma)^2} \right) + \frac{1}{4} \left(\frac{C_{rx}}{1-\gamma} \right)^2 \beta < \frac{1}{2}. \end{cases} \quad (55)$$

Set the step size $\beta = \rho\eta$ with $\rho > 0$. More precisely, we present the parameter configuration to guarantee (55).

$$\begin{cases} \zeta_Q = \zeta_w = 1, \\ \eta < \min \left\{ 1, \frac{(1-\gamma)^2}{16|\mathcal{S}|^2(1+\gamma)^2} \right\}, \\ \rho < \min \left\{ \frac{(1-\gamma)^2}{6|\mathcal{S}|C_{rx}^2}, \frac{(1-\gamma)^2}{8L_w^2} \right\}, \\ \beta < \min \left\{ \frac{\tau^2}{8} \left(L_{\hat{\phi}\pi}^2 + \frac{|\mathcal{S}|^2|\mathcal{A}|^3(1+\gamma)C_{f\pi}^2C_{rx}^2}{(1-\gamma)^2} \right)^{-1}, \frac{1}{4} \left(\frac{L_{\phi}^M}{2} + 2L_w^2 + \frac{1}{4} \left(\frac{C_{rx}}{1-\gamma} \right)^2 \right)^{-1} \right\}, \\ \gamma^{2N} < \frac{1}{8} \left(1 + \frac{4C_{\sigma\pi}^2}{\tau^2} \left(\frac{1}{(1-\gamma)^2} + 4 \right) \right)^{-1}, \end{cases} \quad (56)$$

which means $N \sim \mathcal{O}(1)$ and the step sizes can be chosen as constants. In this way, (54) implies

$$\mathcal{L}_{k+1} - \mathcal{L}_k \leq -\frac{\beta}{2} \|\nabla\phi(x_k)\|_2^2.$$

Summing and telescoping it, we have

$$\frac{1}{K} \sum_{k=0}^K \|\nabla\phi(x_k)\|_2^2 \leq \frac{2\mathcal{L}_0}{K\beta} = \frac{2}{K\beta} (\phi(x_0) + \delta_Q^0 + \delta_w^0) = \mathcal{O}\left(\frac{1}{K}\right).$$

□

H ANALYSIS OF SOBIRL

In this section, we prove the convergence of Algorithm 1, SoBiRL. We outline the proof sketch of Theorem 7.6 here. Firstly, we characterize the distributional drift induced by two different policies π^1, π^2 in Lemma H.1. Then, the Lipschitz property of the hyper-gradient is clarified in Proposition H.6, and the quality of the hyper-gradient estimator is measured in Proposition H.7. Based on these two results, we arrive at Theorem H.8, the convergence analysis of SoBiRL.

The next proposition, generalized from Lemma 1 in (Chakraborty et al., 2024), considers the distribution $\rho((d_1, d_2, \dots, d_I); \pi)$, with each trajectory $d_i = \{(s_h^i, a_h^i)\}_{h=0}^{H-1}$ ($i = 1, 2, \dots, I$) sampled from the trajectory distribution $\rho(d; \pi)$, i.e.,

$$P(d; \pi) = \bar{\rho}(s_0) [\prod_{h=0}^{H-2} \pi(a_h | s_h) \bar{P}(s_{h+1} | s_h, a_h)] \pi(a_{H-1} | s_{H-1}),$$

Lemma H.1. Denote the total variation between distributions by $D_F(\cdot, \cdot)$. For any trajectory tuple (d_1, d_2, \dots, d_I) , with each trajectory holding a finite horizon H , we have

$$D_F(\rho((d_1, d_2, \dots, d_I); \pi^1), \rho((d_1, d_2, \dots, d_I); \pi^2)) \leq \frac{HI}{2} \sqrt{|\mathcal{A}|} \|\pi^1 - \pi^2\|_2. \quad (57)$$

Specifically, for any $x_1, x_2 \in \mathbb{R}^n$, take $\pi^1 = \pi^*(x_1), \pi^2 = \pi^*(x_2)$,

$$D_F(\rho((d_1, d_2, \dots, d_I); \pi^*(x_1)), \rho((d_1, d_2, \dots, d_I); \pi^*(x_2))) \leq \frac{HIL_\pi}{2} \sqrt{|\mathcal{A}|} \|x_1 - x_2\|_2.$$

Proof. Firstly, we focus on the situation $I = 1$, i.e., the one-trajectory distribution $\rho(d; \pi)$. Note that the total variation is the F -divergence induced by $F(x) = \frac{1}{2} |x - 1|$. The triangle inequality of the F -divergence leads to a decomposition:

$$D_F(\rho(d; \pi^1), \rho(d; \pi^2)) \leq D_F(\rho(d; \pi^1), \rho(d; p)) + D_F(\rho(d; p), \rho(d; \pi^2)), \quad (58)$$

with p as a mixed policy executing π^2 at the first time-step of any trajectory and obeying π^1 for subsequent timesteps. In this way, by definition of D_F and p ,

$$\begin{aligned} D_F(\rho(d; \pi^1), \rho(d; p)) &= \sum_d P(d; p) F\left(\frac{P(d; \pi^1)}{P(d; p)}\right) \\ &= \sum_d P(d; p) F\left(\frac{\bar{\rho}(s_0) \pi_{s_0 a_0}^1 \bar{P}(s_1 | s_0, a_0) [\prod_{h=1}^{H-2} \bar{P}(s_{h+1} | s_h, a_h) \pi_{s_h a_h}^1] \pi_{s_{H-1} a_{H-1}}^1}{\bar{\rho}(s_0) \pi_{s_0 a_0}^2 \bar{P}(s_1 | s_0, a_0) [\prod_{h=1}^{H-2} \bar{P}(s_{h+1} | s_h, a_h) \pi_{s_h a_h}^1] \pi_{s_{H-1} a_{H-1}}^1}\right) \\ &= \sum_d P(d; p) F\left(\frac{\pi_{s_0 a_0}^1}{\pi_{s_0 a_0}^2}\right). \end{aligned} \quad (59)$$

From (59), it reveals that for different trajectory tuples, they share the same $F(\cdot)$ term if their initial states-actions pairs (s_0, a_0) are identical, which implies

$$\begin{aligned} D_F(\rho(d; \pi^1), \rho(d; p)) &= \sum_{s_0, a_0} \bar{\rho}(s_0) \pi_{s_0 a_0}^2 F\left(\frac{\pi_{s_0 a_0}^1}{\pi_{s_0 a_0}^2}\right) \\ &= \sum_{s_0} \bar{\rho}(s_0) \sum_{a_0} \pi_{s_0 a_0}^2 F\left(\frac{\pi_{s_0 a_0}^1}{\pi_{s_0 a_0}^2}\right) \\ &= \sum_{s_0} \bar{\rho}(s_0) D_F(\pi_{\cdot | s_0}^1, \pi_{\cdot | s_0}^2). \end{aligned} \quad (60)$$

Then we consider the second term in the triangle inequality decomposition (58).

$$\begin{aligned} D_F(\rho(d; p), \rho(d; \pi^2)) &= \sum_d P(d; \pi^2) F\left(\frac{P(d; p)}{P(d; \pi^2)}\right) \\ &= \sum_d P(d; \pi^2) F\left(\frac{\bar{\rho}(s_0) \pi_{s_0 a_0}^2 \bar{P}(s_1 | s_0, a_0) [\prod_{h=1}^{H-2} \bar{P}(s_{h+1} | s_h, a_h) \pi_{s_h a_h}^1] \pi_{s_{H-1} a_{H-1}}^1}{\bar{\rho}(s_0) \pi_{s_0 a_0}^2 \bar{P}(s_1 | s_0, a_0) [\prod_{h=1}^{H-2} \bar{P}(s_{h+1} | s_h, a_h) \pi_{s_h a_h}^2] \pi_{s_{H-1} a_{H-1}}^2}\right) \end{aligned} \quad (61)$$

$$\begin{aligned} &= \sum_{s_0} \bar{\rho}(s_0) \sum_{a_0} \pi_{s_0 a_0}^2 D_F(\rho(d^{H-1}; \pi^1), \rho(d^{H-1}; \pi^2)) \\ &\leq \sum_{s_0} \bar{\rho}(s_0) \sum_{a_0} \pi_{s_0 a_0}^2 D_F(\rho(d^{H-1}; \pi^1), \rho(d^{H-1}; p')) \\ &\quad + \sum_{s_0} \bar{\rho}(s_0) \sum_{a_0} \pi_{s_0 a_0}^2 D_F(\rho(d^{H-1}; p'), \rho(d^{H-1}; \pi^2)), \end{aligned} \quad (62)$$

where d^{H-1} denotes the trajectory with $H - 1$ horizon, obtained by cutting s_0 and a_0 . One can derive (61) since p aligns with π^2 at the first timestep, and with π^1 at the following steps, and (62) follows from constructing a new mixed policy p' in a similar way and applying the triangle inequality. A subsequent result is that we reduce the original divergence between π^1 and π^2 measured on H -length trajectories to divergence measured on one-length trajectories in (60) and $(H - 1)$ -length trajectories in (62). Repeating the process on the $(H - 1)$ -length

trajectories in (62) yields

$$\begin{aligned}
 & D_F(\rho(d; \pi^1), \rho(d; \pi^2)) \\
 & \leq \sum_{h=0}^{H-1} \mathbb{E}_{s' \sim \rho_h} [D_F(\pi_{\cdot|s'}^1, \pi_{\cdot|s'}^2)] \\
 & \leq H D_F(\pi_{\cdot|s^*}^1, \pi_{\cdot|s^*}^2) \\
 & = \frac{H}{2} \left\| \pi_{\cdot|s^*}^1 - \pi_{\cdot|s^*}^2 \right\|_1 \\
 & \leq \frac{H}{2} \sqrt{|\mathcal{A}|} \|\pi^1 - \pi^2\|_2,
 \end{aligned} \tag{63}$$

where ρ_h is a state distribution related to the timestep h and

$$s^* = \arg \max_{s' \in \mathcal{S}} D_F(\pi_{\cdot|s'}^1, \pi_{\cdot|s'}^2).$$

Note that $\rho((d_1, d_2, \dots, d_I); \pi)$ is the product measure of $\rho(d_i; \pi)$, ($i = 1, 2, \dots, I$), i.e., $\rho((d_1, d_2, \dots, d_I); \pi) = \rho(d_1; \pi) \times \rho(d_2; \pi) \times \dots \times \rho(d_I; \pi)$. By the total variation inequality for the product measure,

$$\begin{aligned}
 D_F(\rho((d_1, d_2, \dots, d_I); \pi^1), \rho((d_1, d_2, \dots, d_I); \pi^2)) & \leq \sum_{i=1}^I D_F(\rho(d_i; \pi^1), \rho(d_i; \pi^2)) \\
 & \leq \frac{HI}{2} \sqrt{|\mathcal{A}|} \|\pi^1 - \pi^2\|_2.
 \end{aligned}$$

□

A byproduct of Lemma H.1 is the lemma presented below, which measures the difference between the expectations of the same function evaluated on different distributions.

Lemma H.2. *If the vector-valued function $z(d_1, d_2, \dots, d_I; x) \in \mathbb{R}^n$ is bounded by C , i.e., $\|z(d_1, d_2, \dots, d_I; x)\|_2 \leq C$, then for any trajectory tuple (d_1, d_2, \dots, d_I) , with each trajectory holding a finite horizon H , and policies π_1, π_2 , it holds*

$$\|\mathbb{E}_{d_i \sim \rho(d; \pi_1)} [z(d_1, d_2, \dots, d_I; x)] - \mathbb{E}_{d_i \sim \rho(d; \pi_2)} [z(d_1, d_2, \dots, d_I; x)]\|_2 \leq CHI \sqrt{|\mathcal{A}|} \|\pi_1 - \pi_2\|_2.$$

Proof. We resort to the property of the Integral Probability Metric (IPM) (Sriperumbudur et al., 2009),

$$\sup_{|\xi| \leq B} \mathbb{E}_{d_i \sim \rho(d; \pi_1)} [\xi] - \mathbb{E}_{d_i \sim \rho(d; \pi_2)} [\xi] = 2B \times D_F(\rho(d; \pi^*(x_1)), \rho(d; \pi^*(x_2))),$$

where D_F is the total variation between distributions. Subsequently, it yields

$$\begin{aligned}
 & |b^\top (\mathbb{E}_{d_i \sim \rho(d; \pi_1)} [z(d_1, d_2, \dots, d_I; x_2)] - \mathbb{E}_{d_i \sim \rho(d; \pi_2)} [z(d_1, d_2, \dots, d_I; x_2)])| \\
 & \leq 2C \|b\|_2 D_F(\rho((d_1, d_2, \dots, d_I); \pi_1), \rho((d_1, d_2, \dots, d_I); \pi_2))
 \end{aligned}$$

for any $b \in \mathbb{R}^n$, which means

$$\begin{aligned}
 & \|\mathbb{E}_{d_i \sim \rho(d; \pi_1)} [z(d_1, d_2, \dots, d_I; x_2)] - \mathbb{E}_{d_i \sim \rho(d; \pi_2)} [z(d_1, d_2, \dots, d_I; x_2)]\|_2 \\
 & \leq 2C \times D_F(\rho((d_1, d_2, \dots, d_I); \pi_1), \rho((d_1, d_2, \dots, d_I); \pi_2)).
 \end{aligned} \tag{64}$$

Combining (57) and (64) completes the proof. □

Drawing from the distributional drift investigated by Lemma H.2, we can bound the difference between expectations, where both functions and distributions are associated with the upper-level variable x .

Lemma H.3. Under Assumption 7.3, if $z(d_1, d_2, \dots, d_I; x) \in \mathbb{R}^n$ is bounded by C , i.e., $\|z(d_1, d_2, \dots, d_I; x)\|_2 \leq C$, and is L_z -Lipschitz continuous with respect to x , then the function

$$Z(x) = \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} [z(d_1, d_2, \dots, d_I; x)]$$

is L_Z -Lipschitz continuous with $L_Z = L_z + CHIL_\pi \sqrt{|\mathcal{A}|}$, i.e., for any $x_1, x_2 \in \mathbb{R}^n$,

$$\|Z(x_1) - Z(x_2)\|_2 \leq L_Z \|x_1 - x_2\|_2.$$

Proof. To begin with, we decompose $Z(x_1) - Z(x_2)$ into two terms,

$$\begin{aligned} Z(x_1) - Z(x_2) &= (\mathbb{E}_{d_i \sim \rho(d; \pi^*(x_1))} [z(d_1, d_2, \dots, d_I; x_1)] - \mathbb{E}_{d_i \sim \rho(d; \pi^*(x_1))} [z(d_1, d_2, \dots, d_I; x_2)]) \\ &\quad + (\mathbb{E}_{d_i \sim \rho(d; \pi^*(x_1))} [z(d_1, d_2, \dots, d_I; x_2)] - \mathbb{E}_{d_i \sim \rho(d; \pi^*(x_2))} [z(d_1, d_2, \dots, d_I; x_2)]). \end{aligned}$$

The first term can be bounded by the Lipschitz continuity of z since the two expectations share the same distribution $\rho(d; \pi^*(x_1))$. To control the second term, applying Lemma H.2 and considering the Lipschitz continuity of $\pi^*(x)$ lead to

$$\left\| \mathbb{E}_{d_i \sim \rho(d; \pi^*(x_1))} [z(d_1, d_2, \dots, d_I; x_2)] - \mathbb{E}_{d_i \sim \rho(d; \pi^*(x_2))} [z(d_1, d_2, \dots, d_I; x_2)] \right\| \leq CHIL_\pi \sqrt{|\mathcal{A}|} \|x_1 - x_2\|_2.$$

It completes the proof by

$$\|Z(x_1) - Z(x_2)\|_2 \leq L_z \|x_1 - x_2\|_2 + CHIL_\pi \sqrt{|\mathcal{A}|} \|x_1 - x_2\|_2.$$

□

The following proposition measures the quality of the implicit differentiation estimators. A similar result is introduced as an assumption in (Shen et al., 2024) (Lemma 18, condition (c)). We claim that it is satisfied in our setting.

Proposition H.4. Under Assumption 7.3, for any upper-level variable $x \in \mathbb{R}^n$ and policies π^1, π^2 , we have

$$\begin{aligned} \max_{s,a} \left\| \tilde{\nabla} Q_{sa}(x, \pi^1) - \tilde{\nabla} Q_{sa}(x, \pi^2) \right\|_2 &\leq \frac{(1+\gamma)C_{rx}\sqrt{|\mathcal{A}|}}{(1-\gamma)^2} \|\pi^1 - \pi^2\|_2, \\ \max_s \left\| \tilde{\nabla} V_s(x, \pi^1) - \tilde{\nabla} V_s(x, \pi^2) \right\|_2 &\leq \frac{(1+\gamma)C_{rx}\sqrt{|\mathcal{A}|}}{(1-\gamma)^2} \|\pi^1 - \pi^2\|_2. \end{aligned}$$

Specifically, taking $\pi^1 = \pi^*(x)$ and $\pi^2 = \pi$ yields

$$\begin{aligned} \max_{s,a} \left\| \nabla Q_{sa}^*(x) - \tilde{\nabla} Q_{sa}(x, \pi) \right\|_2 &\leq \frac{(1+\gamma)C_{rx}\sqrt{|\mathcal{A}|}}{(1-\gamma)^2} \|\pi^*(x) - \pi\|_2 \\ \max_s \left\| \nabla V_s^*(x) - \tilde{\nabla} V_s(x, \pi) \right\|_2 &\leq \frac{(1+\gamma)C_{rx}\sqrt{|\mathcal{A}|}}{(1-\gamma)^2} \|\pi^*(x) - \pi\|_2. \end{aligned}$$

Proof. Recalling the expression of $\tilde{\nabla} V_s(x, \pi)$

$$\tilde{\nabla} V_s(x, \pi) = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \nabla r_{s_t, a_t}(x) | s_0 = s, \pi, \mathcal{M}_\tau(x) \right],$$

we obtain the boundedness

$$\left\| \tilde{\nabla} V_s(x, \pi) \right\|_2 \leq \frac{C_{rx}}{1-\gamma},$$

and the recursive rule

$$\tilde{\nabla} V_s(x, \pi^1) = \sum_a \pi_{sa}^1(x) \nabla r_{sa}(x) + \gamma \mathbb{E}_{s' \sim \rho(s_1; \pi^1)} [\tilde{\nabla} V_{s'}(x, \pi^1)]. \quad (65)$$

$$\tilde{\nabla} V_s(x, \pi^2) = \sum_a \pi_{sa}^2(x) \nabla r_{sa}(x) + \gamma \mathbb{E}_{s' \sim \rho(s_1; \pi^2)} [\tilde{\nabla} V_{s'}(x, \pi^2)]. \quad (66)$$

Subtracting (65) by (66) yields

$$\begin{aligned}\tilde{\nabla}V_s(x, \pi^1) - \tilde{\nabla}V_s(x, \pi^2) &= \sum_a (\pi_{sa}^1(x) - \pi_{sa}^2(x)) \nabla r_{sa}(x) \\ &\quad + \gamma \left(\mathbb{E}_{s' \sim \rho(s_1; \pi^1)} [\tilde{\nabla}V_{s'}(x, \pi^1)] - \mathbb{E}_{s' \sim \rho(s_1; \pi^2)} [\tilde{\nabla}V_{s'}(x, \pi^2)] \right)\end{aligned}\quad (67)$$

The first term can be bounded by applying Lemma H.2 with $I = 1, H = 1, \|\nabla r_{sa}(x)\|_2 \leq C_{rx}$, and the second term can be decomposed into

$$\begin{aligned}&\mathbb{E}_{s' \sim \rho(s_1; \pi^1)} [\tilde{\nabla}V_{s'}(x, \pi^1)] - \mathbb{E}_{s' \sim \rho(s_1; \pi^2)} [\tilde{\nabla}V_{s'}(x, \pi^2)] \\ &= \mathbb{E}_{s' \sim \rho(s_1; \pi^1)} [\tilde{\nabla}V_{s'}(x, \pi^1) - \tilde{\nabla}V_{s'}(x, \pi^2)] \\ &\quad + \mathbb{E}_{s' \sim \rho(s_1; \pi^1)} [\tilde{\nabla}V_{s'}(x, \pi^2)] - \mathbb{E}_{s' \sim \rho(s_1; \pi^2)} [\tilde{\nabla}V_{s'}(x, \pi^2)],\end{aligned}\quad (68)$$

where the term (68) can be bounded by applying Lemma H.2 with $I = 1, H = 2$ and $\|\tilde{\nabla}V_{s'}(x, \pi)\|_2 \leq \frac{C_{rx}}{1-\gamma}$. Defining

$$\Delta V := \max_s \|\tilde{\nabla}V_s(x, \pi^1) - \tilde{\nabla}V_s(x, \pi^2)\|_2,$$

and collecting the observations above into (67) lead to

$$\Delta V \leq C_{rx} \sqrt{|\mathcal{A}|} \|\pi^1 - \pi^2\|_2 + \gamma \left(\Delta V + \frac{2C_{rx} \sqrt{|\mathcal{A}|}}{1-\gamma} \|\pi^1 - \pi^2\|_2 \right),$$

which means

$$\Delta V \leq \frac{(1+\gamma)C_{rx} \sqrt{|\mathcal{A}|}}{(1-\gamma)^2} \|\pi^1 - \pi^2\|_2.$$

By the recursive rule

$$\tilde{\nabla}Q_{sa}(x, \pi) = \nabla r_{sa}(x) + \gamma \mathbb{E}_{s' \sim P(\cdot | sa)} [\tilde{\nabla}V_{s'}(x, \pi)],$$

we achieve the similar property of $\tilde{\nabla}Q(x, \pi)$. \square

A byproduct of Proposition H.4 is the Lipschitz smoothness of the implicit differentiations, $V^*(x)$ and $Q^*(x)$.

Proposition H.5. Under Assumption 7.3, given $x_1, x_2 \in \mathbb{R}^n$, we have for any $s \in \mathcal{S}, a \in \mathcal{A}$,

$$\begin{aligned}\|\nabla Q_{sa}^*(x_1) - \nabla Q_{sa}^*(x_2)\|_2 &\leq L_{V1} \|x_1 - x_2\|_2, \\ \|\nabla V_s^*(x_1) - \nabla V_s^*(x_2)\|_2 &\leq L_{V1} \|x_1 - x_2\|_2,\end{aligned}$$

where $L_{V1} > 0$ is the smoothness constant for the value functions,

$$L_{V1} = \frac{(1+\gamma)C_{rx} L_\pi \sqrt{|\mathcal{A}|}}{(1-\gamma)^2} + \frac{L_r}{1-\gamma}.$$

Proof. Proposition E.1 and the definition of the implicit differentiation reveal that

$$\nabla V_s^*(x) = \tilde{\nabla}V_s(x, \pi^*(x)) = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \nabla r_{s_t, a_t}(x) | s_0 = s, \pi^*(x), \mathcal{M}_\tau(x) \right].$$

Through this equality, the following decomposition is derived,

$$\begin{aligned}\nabla V_s^*(x_1) - \nabla V_s^*(x_2) &= \left(\tilde{\nabla}V_s(x_1, \pi^*(x_1)) - \tilde{\nabla}V_s(x_1, \pi^*(x_2)) \right) \\ &\quad + \left(\tilde{\nabla}V_s(x_1, \pi^*(x_2)) - \tilde{\nabla}V_s(x_2, \pi^*(x_2)) \right).\end{aligned}$$

Apply Proposition H.4 and Lipschitz continuity of π^* ,

$$\begin{aligned} \left\| \tilde{\nabla}V_s(x_1, \pi^*(x_1)) - \tilde{\nabla}V_s(x_1, \pi^*(x_2)) \right\|_2 &\leq \frac{(1+\gamma)C_{rx}\sqrt{|\mathcal{A}|}}{(1-\gamma)^2} \|\pi^*(x_1) - \pi^*(x_2)\|_2 \\ &\leq \frac{(1+\gamma)C_{rx}L_\pi\sqrt{|\mathcal{A}|}}{(1-\gamma)^2} \|x_1 - x_2\|_2. \end{aligned}$$

Additionally, Lipschitz smoothness of r yields

$$\begin{aligned} &\left\| \tilde{\nabla}V_s(x_1, \pi^*(x_2)) - \tilde{\nabla}V_s(x_2, \pi^*(x_2)) \right\|_2 \\ &\leq \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \|\nabla r_{s_t, a_t}(x_1) - \nabla r_{s_t, a_t}(x_2)\|_2 \mid s_0 = s, \pi^*(x_2) \right] \\ &\leq \frac{L_r}{1-\gamma} \|x_1 - x_2\|_2. \end{aligned}$$

Consequently,

$$\|\nabla V_s^*(x_1) - \nabla V_s^*(x_2)\|_2 \leq \frac{(1+\gamma)C_{rx}L_\pi\sqrt{|\mathcal{A}|}}{(1-\gamma)^2} \|x_1 - x_2\|_2 + \frac{L_r}{1-\gamma} \|x_1 - x_2\|_2 = L_{V1} \|x_1 - x_2\|_2.$$

In a similar fashion, we can also conclude

$$\|\nabla Q_{sa}^*(x_1) - \nabla Q_{sa}^*(x_2)\|_2 \leq L_{V1} \|x_1 - x_2\|_2.$$

□

In this way, it establishes the Lipschitz smoothness of the hyper-objective $\phi(x)$, which plays a key role in the convergence analysis.

Proposition H.6. *Under the Assumptions 7.2 and 7.3, $\nabla\phi(x)$ is L_ϕ -Lipschitz continuous, i.e., for any $x_1, x_2 \in \mathbb{R}^n$,*

$$\|\nabla\phi(x_1) - \nabla\phi(x_2)\|_2 \leq L_\phi \|x_1 - x_2\|_2,$$

where $L_\phi > 0$ is defined by

$$L_\phi = L_{l1} + L_l H I L_\pi \sqrt{|\mathcal{A}|} + 2\tau^{-1} H I \left(C_l L_{V1} + \frac{C_{rx}}{1-\gamma} L_l \right) + \frac{2H^2 I^2 C_l C_{rx} L_\pi}{\tau(1-\gamma)} \sqrt{|\mathcal{A}|}.$$

Proof. Given

$$\begin{aligned} \nabla\phi(x) &= \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} [\nabla l(d_1, d_2, \dots, d_I; x)] \\ &\quad + \tau^{-1} \mathbb{E}_{d_i \sim \rho(d; \pi^*(x))} \left[l(d_1, d_2, \dots, d_I; x) \left(\sum_i \sum_h \nabla (Q_{s_h^i a_h^i}^*(x) - V_{s_h^i}^*(x)) \right) \right], \end{aligned} \tag{69}$$

it is deduced from Lemma H.3 that the first term in (69) is $(L_{l1} + L_l H I L_\pi \sqrt{|\mathcal{A}|})$ -Lipschitz continuous since $\nabla l(d_1, d_2, \dots, d_I; x)$ is L_l -bounded and L_{l1} -Lipschitz continuous, revealed by Assumption 7.2. Subsequently, we consider the second term in (69). Specifically, assembling the L_l -Lipschitz continuity of l , the L_{V1} -Lipschitz continuity of ∇V_s^* , and the L_{V1} -Lipschitz continuity of ∇Q_{sa}^* , along with the boundedness of these functions, we conclude the function defined by

$$z(d_1, d_2, \dots, d_I; x) := l(d_1, d_2, \dots, d_I; x) \left(\sum_i \sum_h \nabla (Q_{s_h^i a_h^i}^*(x) - V_{s_h^i}^*(x)) \right)$$

is bounded by

$$\|z(d_1, d_2, \dots, d_I; x)\|_2 \leq 2HIC_l \frac{C_{rx}}{1-\gamma},$$

and L_z -Lipschitz continuous with

$$L_z = 2HI \left(C_l L_{V_1} + \frac{C_{rx}}{1-\gamma} L_l \right).$$

Applying Lemma H.3 on the second term in (69) implies it is $\left(L_z + \frac{2H^2 I^2 C_l C_{rx} L_\pi}{1-\gamma} \sqrt{|\mathcal{A}|} \right)$ -Lipschitz continuous. Combining the Lipschitz constants of the two terms in (69) provides the Lipschitz constant of $\nabla\phi(x)$, i.e.,

$$\begin{aligned} L_\phi &= \left(L_{l1} + L_l HIL_\pi \sqrt{|\mathcal{A}|} \right) + \tau^{-1} \left(L_z + \frac{2H^2 I^2 C_l C_{rx} L_\pi}{1-\gamma} \sqrt{|\mathcal{A}|} \right) \\ &= L_{l1} + L_l HIL_\pi \sqrt{|\mathcal{A}|} + 2\tau^{-1} HI \left(C_l L_{V_1} + \frac{C_{rx}}{1-\gamma} L_l \right) + \frac{2H^2 I^2 C_l C_{rx} L_\pi}{\tau(1-\gamma)} \sqrt{|\mathcal{A}|}. \end{aligned}$$

□

Next, we analyze the error brought by the estimator $\tilde{\nabla}\phi(x, \pi)$ to approximate the true hyper-gradient $\nabla\phi(x)$ in the following proposition. $\tilde{\nabla}\phi(x, \pi)$.

Proposition H.7. *Under Assumptions 7.2 and 7.3, for any upper-level variable $x \in \mathbb{R}^n$ and policies π^1, π^2 , we have*

$$\left\| \tilde{\nabla}\phi(x, \pi^1) - \tilde{\nabla}\phi(x, \pi^2) \right\|_2 \leq L_{\tilde{\phi}} \left\| \pi^1 - \pi^2 \right\|_2,$$

with $L_{\tilde{\phi}} = HIL_l \sqrt{|\mathcal{A}|} + \frac{2\tau^{-1} C_l C_{rx} H I \sqrt{|\mathcal{A}|}}{1-\gamma} \left(\frac{1+\gamma}{1-\gamma} + HI \right)$. Specifically, taking $\pi^1 = \pi^*(x)$ and $\pi^2 = \pi$ yields

$$\left\| \nabla\phi(x) - \tilde{\nabla}\phi(x, \pi) \right\|_2 \leq L_{\tilde{\phi}} \left\| \pi^*(x) - \pi \right\|_2.$$

Proof. Considering $\|\nabla l\|_2 \leq L_l$ and applying Lemma H.2, we obtain

$$\left\| \mathbb{E}_{d_i \sim \rho(d; \pi^1)} [\nabla l(d_1, d_2, \dots, d_I; x)] - \mathbb{E}_{d_i \sim \rho(d; \pi^2)} [\nabla l(d_1, d_2, \dots, d_I; x)] \right\|_2 \leq HIL_l \sqrt{|\mathcal{A}|} \left\| \pi^1 - \pi^2 \right\|_2.$$

Denote $z(d_1, d_2, \dots, d_I; x, \pi) := l(d_1, d_2, \dots, d_I; x) \left(\sum_i \sum_h \tilde{\nabla} (Q_{s_h^i a_h^i} - V_{s_h^i}) (x, \pi) \right)$. Taking into account the boundedness and Lipschitz continuity of l in Assumption 7.2, and of $\tilde{\nabla} Q$, $\tilde{\nabla} V$ in Proposition H.4, we can establish the boundedness,

$$\|z(d_1, d_2, \dots, d_I; x, \pi)\|_2 \leq 2C_l HI \frac{C_{rx}}{1-\gamma}, \quad (70)$$

and the Lipschitz continuity of $z(d_1, d_2, \dots, d_I; x, \pi)$, i.e.,

$$\|z(d_1, d_2, \dots, d_I; x, \pi_1) - z(d_1, d_2, \dots, d_I; x, \pi_2)\|_2 \leq 2C_l HI \frac{(1+\gamma)C_{rx}\sqrt{|\mathcal{A}|}}{(1-\gamma)^2} \left\| \pi_1 - \pi_2 \right\|_2. \quad (71)$$

In this way,

$$\begin{aligned} &\left\| \mathbb{E}_{d_i \sim \rho(d; \pi^1)} [z(x, \pi_1)] - \mathbb{E}_{d_i \sim \rho(d; \pi^2)} [z(x, \pi_2)] \right\|_2 \\ &\leq \left\| \mathbb{E}_{d_i \sim \rho(d; \pi^1)} [z(x, \pi_1)] - \mathbb{E}_{d_i \sim \rho(d; \pi^1)} [z(x, \pi_2)] \right\|_2 \end{aligned} \quad (72)$$

$$+ \left\| \mathbb{E}_{d_i \sim \rho(d; \pi^1)} [z(x, \pi_2)] - \mathbb{E}_{d_i \sim \rho(d; \pi^2)} [z(x, \pi_2)] \right\|_2 \quad (73)$$

$$\begin{aligned} &\leq 2C_l HI \frac{(1+\gamma)C_{rx}\sqrt{|\mathcal{A}|}}{(1-\gamma)^2} \left\| \pi_1 - \pi_2 \right\|_2 + 2C_l H^2 I^2 \frac{C_{rx}}{1-\gamma} \sqrt{|\mathcal{A}|} \left\| \pi_1 - \pi_2 \right\|_2, \\ &= \frac{2C_l C_{rx} HI \sqrt{|\mathcal{A}|}}{1-\gamma} \left(\frac{1+\gamma}{1-\gamma} + HI \right) \left\| \pi_1 - \pi_2 \right\|_2, \end{aligned}$$

where we bound (72) by the Lipschitz continuity (71), and bound (73) by applying Lemma H.2 with (70). Given

$$\begin{aligned} \tilde{\nabla}\phi(x, \pi) &= \mathbb{E}_{d_i \sim \rho(d; \pi)} [\nabla l(d_1, d_2, \dots, d_I; x)] \\ &+ \tau^{-1} \mathbb{E}_{d_i \sim \rho(d; \pi)} \left[l(d_1, d_2, \dots, d_I; x) \left(\sum_i \sum_h \tilde{\nabla} (Q_{s_h^i a_h^i} - V_{s_h^i}) (x, \pi) \right) \right], \end{aligned}$$

we have

$$\begin{aligned}
 & \left\| \tilde{\nabla} \phi(x, \pi^1) - \tilde{\nabla} \phi(x, \pi^2) \right\|_2 \\
 & \leq \left\| \mathbb{E}_{d_i \sim \rho(d; \pi^1)} [\nabla l(d_1, d_2, \dots, d_I; x)] - \mathbb{E}_{d_i \sim \rho(d; \pi^2)} [\nabla l(d_1, d_2, \dots, d_I; x)] \right\|_2 \\
 & \quad + \tau^{-1} \left\| \mathbb{E}_{d_i \sim \rho(d; \pi^1)} [z(d_1, d_2, \dots, d_I; x, \pi^1)] - \mathbb{E}_{d_i \sim \rho(d; \pi^2)} [z(d_1, d_2, \dots, d_I; x, \pi^2)] \right\|_2 \\
 & \leq HIL_l \sqrt{|\mathcal{A}|} \left\| \pi^1 - \pi^2 \right\|_2 + \frac{2\tau^{-1} C_l C_{rx} H I \sqrt{|\mathcal{A}|}}{1-\gamma} \left(\frac{1+\gamma}{1-\gamma} + HI \right) \left\| \pi_1 - \pi_2 \right\|_2 \\
 & = L_{\tilde{\phi}} \left\| \pi_1 - \pi_2 \right\|_2.
 \end{aligned}$$

□

With all the lemmas and propositions now proven, we turn our attention to presenting the convergence theorem for the proposed algorithm, SoBiRL.

Theorem H.8. *Under Assumptions 7.2 and 7.3 and given the accuracy in Algorithm 1, we can set the constant step size $\beta < \frac{1}{2L_\phi}$, then the iterates $\{x_k\}$ satisfy*

$$\frac{1}{K} \sum_{k=1}^K \left\| \nabla \phi(x_k) \right\|_2^2 \leq \frac{\phi(x_1) - \phi^*}{K \left(\frac{\beta}{2} - \beta^2 L_\phi \right)} + \frac{1+2\beta L_\phi}{1-2\beta L_\phi} L_{\tilde{\phi}}^2 \epsilon,$$

where ϕ^* is the minimum of the hyper-objective $\phi(x)$, and

$$\begin{aligned}
 L_\phi &= L_{l1} + L_l H I L_\pi \sqrt{|\mathcal{A}|} + 2\tau^{-1} H I \left(C_l L_{V_1} + \frac{C_{rx}}{1-\gamma} L_l \right) + \frac{2H^2 I^2 C_l C_{rx} L_\pi}{\tau(1-\gamma)} \sqrt{|\mathcal{A}|}, \\
 L_{\tilde{\phi}} &= H I L_l \sqrt{|\mathcal{A}|} + \frac{2\tau^{-1} C_l C_{rx} H I \sqrt{|\mathcal{A}|}}{1-\gamma} \left(\frac{1+\gamma}{1-\gamma} + HI \right).
 \end{aligned}$$

Proof. In this proof, all the notation $\|\cdot\|$ without subscripts denotes the 2-norm for simplicity. Based on the L_ϕ -Lipschitz smoothness of $\phi(x)$ revealed by Proposition H.6, a gradient descent step results in the decrease in the hyper-objective φ :

$$\begin{aligned}
 \phi(x_{k+1}) &\leq \phi(x_k) + \langle \nabla \phi(x_k), x_{k+1} - x_k \rangle + \frac{L_\phi}{2} \|x_{k+1} - x_k\|^2 \\
 &= \phi(x_k) - \beta \left\langle \nabla \phi(x_k), \tilde{\nabla} \phi(x_k, \pi_k) \right\rangle + \frac{\beta^2 L_\phi}{2} \left\| \tilde{\nabla} \phi(x_k, \pi_k) \right\|^2
 \end{aligned} \tag{74}$$

$$\begin{aligned}
 &= \phi(x_k) - \beta \left\langle \nabla \phi(x_k), \tilde{\nabla} \phi(x_k, \pi_k) - \nabla \phi(x_k) + \nabla \phi(x_k) \right\rangle + \frac{\beta^2 L_\phi}{2} \left\| \tilde{\nabla} \phi(x_k, \pi_k) \right\|^2 \\
 &= \phi(x_k) - \beta \left\| \nabla \phi(x_k) \right\|^2 - \beta \left\langle \nabla \phi(x_k), \tilde{\nabla} \phi(x_k, \pi_k) - \nabla \phi(x_k) \right\rangle \\
 &\quad + \frac{\beta^2 L_\phi}{2} \left\| \tilde{\nabla} \phi(x_k, \pi_k) - \nabla \phi(x_k) + \nabla \phi(x_k) \right\|^2 \\
 &\leq \phi(x_k) - \beta \left\| \nabla \phi(x_k) \right\|^2 + \frac{\beta}{2} \left(\left\| \tilde{\nabla} \phi(x_k, \pi_k) - \nabla \phi(x_k) \right\|^2 + \left\| \nabla \phi(x_k) \right\|^2 \right)
 \end{aligned} \tag{75}$$

$$\begin{aligned}
 &\quad + \beta^2 L_\phi \left(\left\| \tilde{\nabla} \phi(x_k, \pi_k) - \nabla \phi(x_k) \right\|^2 + \left\| \nabla \phi(x_k) \right\|^2 \right) \\
 &= \phi(x_k) - \left(\frac{\beta}{2} - \beta^2 L_\phi \right) \left\| \nabla \phi(x_k) \right\|^2 + \left(\frac{\beta}{2} + \beta^2 L_\phi \right) \left\| \nabla \phi(x_k) - \tilde{\nabla} \phi(x_k, \pi_k) \right\|^2,
 \end{aligned} \tag{76}$$

where the updating rule $x_{k+1} = x_k - \beta \tilde{\nabla} \phi(x_k, \pi_k)$ yields (74), and (75) comes from the inequalities $|\langle a, b \rangle| \leq \frac{1}{2} (\|a\|^2 + \|b\|^2)$ and $\frac{1}{2} \|a+b\|^2 \leq \|a\|^2 + \|b\|^2$. Drawing from Proposition H.7 and the approximate lower-level solution π_k satisfying $\|\pi^*(x_k) - \pi_k\|^2 \leq \epsilon$, we can incorporate

$$\left\| \nabla \phi(x_k) - \tilde{\nabla} \phi(x_k, \pi_k) \right\|^2 \leq L_{\tilde{\phi}}^2 \left\| \pi^*(x_k) - \pi_k \right\|^2 \leq L_{\tilde{\phi}}^2 \epsilon,$$

into (76) and gain the estimation

$$\phi(x_{k+1}) \leq \phi(x_k) - \left(\frac{\beta}{2} - \beta^2 L_\phi \right) \|\nabla \phi(x_k)\|^2 + \left(\frac{\beta}{2} + \beta^2 L_\phi \right) L_\phi^2 \epsilon. \quad (77)$$

Telescoping index from $k = 1$ to $k = K$, we find that

$$\left(\frac{\beta}{2} - \beta^2 L_\phi \right) \sum_{k=1}^K \|\nabla \phi(x_k)\|^2 \leq \phi(x_1) - \phi^* + K \left(\frac{\beta}{2} + \beta^2 L_\phi \right) L_\phi^2 \epsilon,$$

where we define ϕ^* to be minimum of the hyper-objective. By setting the step size $\beta < \frac{1}{2L_\phi}$ and dividing both sides by $K \left(\frac{\beta}{2} - \beta^2 L_\phi \right)$, we arrive at the conclusion. \square

The proof of Theorem H.8 reveals that $\frac{1}{K} \sum \|\nabla \phi(x_k)\|_2^2 = \mathcal{O}(\frac{1}{K\beta})$, and we can substitute $\beta = \mathcal{O}(1/L_\phi)$ and $L_\phi = \mathcal{O}(\sqrt{|\mathcal{A}|})$ to obtain the outer iteration complexity as $\mathcal{O}(\sqrt{|\mathcal{A}|}\epsilon^{-1})$.

I STOCHASTIC MODEL-FREE SOFT BILEVEL RL

In this section, the details of the stochastic model-free soft bilevel reinforcement learning algorithm, Stoc-SoBiRL are given. Moreover, we provide the statistical properties (bias and variance of the stochastic hyper-gradient) for the sampling scheme Algorithm 3 in Appendix I.2, and the convergence result of Stoc-SoBiRL Algorithm 4 in Appendix I.3.

I.1 Details of Stoc-SoBiRL

The core principle for estimating the hyper-gradient is to sample independent trajectories, evaluate the relevant quantities along each trajectory, and then average these values. Specifically, in the k -th outer iteration, to estimate $\nabla \phi$, an expectation with respect to the random variable \mathbf{d} , we can generate M independent tuples $\mathbf{d}^m := (d_1^m, d_2^m, \dots, d_I^m)$ for $m = 1, 2, \dots, M$ and denote $\mathbf{D} := \{\mathbf{d}^1, \mathbf{d}^2, \dots, \mathbf{d}^M\}$. The next focus is to tackle the terms ∇Q^* and ∇V^* in (13). Assuming access to a generative model (Kearns and Singh, 1998; Li et al., 2024a), from any initial state-action pair (s, a) , we sample J independent trajectories of length T by implementing π_k , i.e., for $j = 1, 2, \dots, J$,

$$\xi_k^j(s, a) := \{(s_0^j = s, a_0^j = a); (s_1^j, a_1^j), \dots, (s_{T-1}^j, a_{T-1}^j)\}. \quad (78)$$

Collecting all these random variables yields $\boldsymbol{\xi}_k := \{\xi_k^j(s, a), j = 1, \dots, J, s \in \mathcal{S}, a \in \mathcal{A}\}$, and the estimator for ∇Q^* can be constructed as

$$\bar{\nabla} Q_{sa}^{\boldsymbol{\xi}_k}(x_k, \pi_k) := \frac{1}{J} \sum_{j=1}^J \sum_{t=0}^{T-1} \gamma^t \nabla r_{s_t^j a_t^j}(x_k). \quad (79)$$

Similarly, the constructions of the random variable $\boldsymbol{\zeta}_k$ and the associated estimator $\bar{\nabla} V^{\boldsymbol{\zeta}_k}$ for ∇V^* are detailed in Algorithm 3. Consequently, we obtain the stochastic hyper-gradient:

$$\bar{\nabla} \phi(\mathbf{D}_k, \boldsymbol{\xi}_k, \boldsymbol{\zeta}_k; x_k, \pi_k) = \frac{1}{M} \sum_{m=1}^M \nabla l(\mathbf{d}_k^m; x_k) + \frac{1}{\tau M} \sum_{m=1}^M l(\mathbf{d}_k^m; x_k) \sum_i \sum_h \bar{\nabla} \left(Q_{s_h^m a_h^m}^{\boldsymbol{\xi}_k} - V_{s_h^m}^{\boldsymbol{\zeta}_k} \right)(x_k, \pi_k), \quad (80)$$

which is abbreviated as $\bar{\nabla} \phi_k$ in the following discussion.

Momentum techniques are known to be beneficial for reducing variance and accelerating algorithms (Cutkosky and Orabona, 2019). To this end, we maintain a momentum-instructed h_k in the k -th outer iteration,

$$h_k = \bar{\nabla} \phi_k + (1 - \mu_k)(h_{k-1} - \bar{\nabla} \phi(\mathbf{D}_k, \boldsymbol{\xi}_k, \boldsymbol{\zeta}_k; x_{k-1}, \pi_{k-1})), \quad (81)$$

which tracks $\bar{\nabla} \phi_k$ via current and historical hyper-gradient estimates. Using h_k to update the upper-level x_k , we obtain the stochastic Algorithm 4, Stoc-SoBiRL.

Algorithm 3 HyperEstimator(x, π, M, J, T)

Input: reward parameter x , policy π , sampling configurations M, J, T

- 1: Independently sample M trajectory tuples $\mathbf{d}^m := (d_1^m, d_2^m, \dots, d_I^m)$ for $m = 1, 2, \dots, M$, with each H -length trajectory $d_i^m = \{(s_h^{m,i}, a_h^{m,i})\}_{h=0}^{H-1}$, and denote $\mathbf{D} := \{\mathbf{d}^1, \mathbf{d}^2, \dots, \mathbf{d}^M\}$
- 2: For $m = 1, 2, \dots, M$, $i = 1, 2, \dots, I$ and $h = 0, 1, \dots, H - 1$, sample $2J$ independent trajectories of length T , i.e., for $j = 1, 2, \dots, J$,

$$\begin{aligned}\xi^j(s_h^{m,i}, a_h^{m,i}) &:= \left\{ \left(s_{q,0}^{m,i,h,j} = s_h^{m,i}, a_{q,0}^{m,i,h,j} = a_h^{m,i} \right); \left(s_{q,1}^{m,i,h,j}, a_{q,1}^{m,i,h,j} \right), \dots, \left(s_{q,T-1}^{m,i,h,j}, a_{q,T-1}^{m,i,h,j} \right) \right\} \\ \zeta^j(s_h^{m,i}) &:= \left\{ \left(s_{v,0}^{m,i,h,j} = s_h^{m,i}; a_{v,0}^{m,i,h,j} \right), \left(s_{v,1}^{m,i,h,j}, a_{v,1}^{m,i,h,j} \right), \dots, \left(s_{v,T-1}^{m,i,h,j}, a_{v,T-1}^{m,i,h,j} \right) \right\}\end{aligned}$$

denote $\boldsymbol{\xi} := \{\xi^j(s_h^{m,i}, a_h^{m,i}), m = 1, \dots, M, j = 1, \dots, J, i = 1, \dots, I, h = 0, \dots, H - 1\}$,
 $\boldsymbol{\zeta} := \{\zeta^j(s_h^{m,i}), m = 1, \dots, M, j = 1, \dots, J, i = 1, \dots, I, h = 0, \dots, H - 1\}$ and compute

$$\begin{aligned}\bar{\nabla}Q_{s_h^{m,i} a_h^{m,i}}^{\boldsymbol{\xi}}(x, \pi) &:= \frac{1}{J} \sum_{j=1}^J \sum_{t=0}^{T-1} \gamma^t \nabla r_{s_{q,t}^{m,i,h,j} a_{q,t}^{m,i,h,j}}(x) \\ \bar{\nabla}V_{s_h^{m,i}}^{\boldsymbol{\zeta}}(x, \pi) &:= \frac{1}{J} \sum_{j=1}^J \sum_{t=0}^{T-1} \gamma^t \nabla r_{s_{v,t}^{m,i,h,j} a_{v,t}^{m,i,h,j}}(x)\end{aligned}$$

- 3: Compute the stochastic hyper-gradient

$$\bar{\nabla}\phi(\mathbf{D}, \boldsymbol{\xi}, \boldsymbol{\zeta}; x, \pi) = \frac{1}{M} \sum_{m=1}^M \nabla l(\mathbf{d}^m; x) + \tau^{-1} \frac{1}{M} \sum_{m=1}^M l(\mathbf{d}^m; x) \sum_i \sum_h \bar{\nabla} \left(Q_{s_h^{m,i} a_h^{m,i}}^{\boldsymbol{\xi}} - V_{s_h^{m,i}}^{\boldsymbol{\zeta}} \right)(x, \pi)$$

Output: $\bar{\nabla}\phi(\mathbf{D}, \boldsymbol{\xi}, \boldsymbol{\zeta}; x, \pi)$, $(\mathbf{D}, \boldsymbol{\xi}, \boldsymbol{\zeta})$
Algorithm 4 Stochastic model-free soft bilevel reinforcement learning algorithm (Stoch-SoBiRL)

Input: iteration number K , step sizes $\{\beta_k, \mu_k\}_{k=1}^K$, initialization x_1, π_0, h_0 , sampling configurations M, J, T , lower-level accuracy $\{\epsilon_k\}_{k=1}^K$

- 1: **for** $k = 1, \dots, K$ **do**
- 2: Solve the lower-level problem with the initialization π_{k-1} to get π_k satisfying $\|\pi_k - \pi^*(x_k)\|_2^2 \leq \epsilon_k$
- 3: Obtain the stochastic hyper-gradient and samples: $\bar{\nabla}\phi_k, (\mathbf{D}_k, \boldsymbol{\xi}_k, \boldsymbol{\zeta}_k) = \text{HyperEstimator}(x_k, \pi_k, M, J, T)$
- 4: Compute the momentum-instructed gradient estimator h_k via (81)
- 5: Implement a stochastic hyper-gradient step $x_{k+1} = x_k - \beta_k h_k$
- 6: **end for**
- 7: Sample \bar{K} from the uniform distribution $\mathcal{U}\{1, 2, \dots, K\}$

Output: $(x_{\bar{K}}, \pi_{\bar{K}})$

I.2 Stochasticity of the Estimator $\bar{\nabla}\phi$

This part analyzes the bias and variance introduced by the stochastic hyper-gradient $\bar{\nabla}\phi$, which serve as a cornerstone for the convergence analysis of the developed Stoc-SoBiRL. To begin with, we show the bias of the estimators $\bar{\nabla}Q^{\xi_k}$ and $\bar{\nabla}V^{\zeta_k}$ to $\tilde{\nabla}Q$ and $\tilde{\nabla}V$, respectively.

Lemma I.1. *Under Assumption 7.3, we have that given any $s \in \mathcal{S}$ and $a \in \mathcal{A}$,*

$$\begin{aligned} \left\| \mathbb{E}_{\xi_k} [\bar{\nabla}Q_{sa}^{\xi_k}(x_k, \pi_k)] - \tilde{\nabla}Q_{sa}(x_k, \pi_k) \right\|_2 &\leq \frac{\gamma^T C_{rx}}{1-\gamma}, \\ \left\| \mathbb{E}_{\zeta_k} [\bar{\nabla}V_s^{\zeta_k}(x_k, \pi_k)] - \tilde{\nabla}V_s(x_k, \pi_k) \right\|_2 &\leq \frac{\gamma^T C_{rx}}{1-\gamma}. \end{aligned} \quad (82)$$

Proof. Recall the expression of $\tilde{\nabla}Q$ in Section 5,

$$\tilde{\nabla}Q_{sa}(x_k, \pi_k) = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \nabla r_{s_t, a_t}(x_k) \mid s_0 = s, a_0 = a, \pi_k \right].$$

By definition of $\bar{\nabla}Q_{sa}^{\xi_k}(x_k, \pi_k)$ in (79), which is indeed a T -depth truncation of $\tilde{\nabla}Q_{sa}(x_k, \pi_k)$, we obtain

$$\mathbb{E}_{\xi_k} [\bar{\nabla}Q_{sa}^{\xi_k}(x_k, \pi_k)] - \tilde{\nabla}Q_{sa}(x_k, \pi_k) = \mathbb{E} \left[\sum_{t=T}^{\infty} \gamma^t \nabla r_{s_t, a_t}(x_k) \mid s_0 = s, a_0 = a, \pi_k \right],$$

which together with the boundedness of ∇r implies

$$\left\| \mathbb{E}_{\xi_k} [\bar{\nabla}Q_{sa}^{\xi_k}(x_k, \pi_k)] - \tilde{\nabla}Q_{sa}(x_k, \pi_k) \right\|_2 \leq \frac{\gamma^T C_{rx}}{1-\gamma}.$$

In a similar fashion, we can also derive the boundedness for the bias of $\bar{\nabla}V_s^{\zeta_k}(x_k, \pi_k)$. \square

Now, we can bound the variance of $\bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; x_k, \pi_k)$ and the bias between it and the targeted $\tilde{\nabla}\phi$.

Proposition I.2. *Under Assumptions 7.2 and 7.3, we have*

$$\left\| \mathbb{E}_{\mathbf{D}_k, \xi_k, \zeta_k} [\bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; x_k, \pi_k)] - \tilde{\nabla}\phi(x_k, \pi_k) \right\|_2 \leq \frac{2\gamma^T HIC_l C_{rx}}{\tau(1-\gamma)}. \quad (83)$$

Additionally, it holds that

$$\mathbb{E}_{\mathbf{D}_k, \xi_k, \zeta_k} \left[\left\| \bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; x_k, \pi_k) - \mathbb{E}_{\mathbf{D}_k, \xi_k, \zeta_k} [\bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; x_k, \pi_k)] \right\|_2^2 \right] \leq \frac{4}{M} \left(L_l + \frac{2HIC_l C_{rx} (1-\gamma^{T_k})}{\tau(1-\gamma)} \right)^2 \quad (84)$$

Proof. Recapping the formulation of $\tilde{\nabla}\phi$ in Section 5,

$$\tilde{\nabla}\phi(x_k, \pi_k) = \mathbb{E}_{\mathbf{d}} [\nabla l(\mathbf{d}; x_k)] + \frac{1}{\tau} \mathbb{E}_{\mathbf{d}} \left[l(\mathbf{d}; x_k) \left(\sum_i \sum_h \tilde{\nabla} (Q_{s_h^i a_h^i} - V_{s_h^i})(x_k, \pi_k) \right) \right],$$

and taking into account the expression of $\bar{\nabla}\phi_k$ in (80), we can compute

$$\begin{aligned} &\mathbb{E}_{\mathbf{D}_k, \xi_k, \zeta_k} [\bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; x_k, \pi_k)] - \tilde{\nabla}\phi(x_k, \pi_k) \\ &= \frac{1}{\tau M} \sum_{m=1}^M \mathbb{E}_{\mathbf{d}^m} \left[l(\mathbf{d}^m; x_k) \sum_i \sum_h \left\{ \mathbb{E}_{\xi_k, \zeta_k} \left[\bar{\nabla} (Q_{s_h^i a_h^i}^{\xi_k} - V_{s_h^i}^{\zeta_k})(x_k, \pi_k) \right] - \tilde{\nabla} (Q_{s_h^i a_h^i} - V_{s_h^i})(x_k, \pi_k) \right\} \right]. \end{aligned}$$

Applying boundedness of l and (82) on the above equation yields (83). The next step is to investigate the variance of the stochastic hyper-gradient. Note that $\bar{\nabla}\phi_k$ is the average of M i.i.d. random variables

$$Z^m := \nabla l(\mathbf{d}^m; x_k) + \frac{1}{\tau} l(\mathbf{d}^m; x_k) \sum_i \sum_h \bar{\nabla} \left(Q_{s_h^{m,i} a_h^{m,i}}^{\xi_k} - V_{s_h^{m,i}}^{\zeta_k} \right) (x_k, \pi_k) \text{ for } m = 1, 2, \dots, M.$$

Considering the boundedness of ∇l , $\bar{\nabla}Q^\xi$ and $\bar{\nabla}V^\zeta$, we obtain

$$\|Z^m - \mathbb{E}[Z^m]\|_2^2 \leq 4 \left(L_l + \frac{2HIC_l C_{rx} (1 - \gamma^{T_k})}{\tau (1 - \gamma)} \right)^2.$$

This reveals

$$\mathbb{E} \left[\left\| \frac{1}{M} \sum_{m=1}^M (Z^m - \mathbb{E}[Z^m]) \right\|_2^2 \right] = \frac{1}{M^2} \sum_{m=1}^M \mathbb{E} \left[\|Z^m - \mathbb{E}[Z^m]\|_2^2 \right] \leq \frac{4}{M} \left(L_l + \frac{2HIC_l C_{rx} (1 - \gamma^{T_k})}{\tau (1 - \gamma)} \right)^2.$$

Noticing that $\bar{\nabla}\phi(\mathbf{D}_k, \boldsymbol{\xi}_k, \boldsymbol{\zeta}_k; x_k, \pi_k) = \frac{1}{M} \sum_{m=1}^M Z^m$ completes the proof. \square

The above proposition means the bias and variance of the stochastic hyper-gradient can be sufficiently small by adjusting the sampling configurations M, J, T .

I.3 Convergence Analysis of Stoc-SoBiRL

In this part, we give a detailed convergence analysis for Stoc-SoBiRL. The Lipschitz properties of $\bar{\nabla}Q^\xi$, $\bar{\nabla}V$ and $\bar{\nabla}\phi$ developed in the following lemmas serve as a foundation for convergence results of Stoc-SoBiRL.

Lemma I.3. *Under Assumption 7.3, for arbitrary upper-level variables $x^1, x^2 \in \mathbb{R}^n$, policies π^1, π^2 , and $s \in \mathcal{S}$, $a \in \mathcal{A}$, we have*

$$\begin{aligned} \|\bar{\nabla}Q_{sa}^\xi(x^1, \pi^1) - \bar{\nabla}Q_{sa}^\xi(x^2, \pi^2)\|_2 &\leq \frac{(1 - \gamma^T) L_r}{1 - \gamma} \|x^1 - x^2\|_2, \\ \|\bar{\nabla}V_s^\zeta(x^1, \pi^1) - \bar{\nabla}V_s^\zeta(x^2, \pi^2)\|_2 &\leq \frac{(1 - \gamma^T) L_r}{1 - \gamma} \|x^1 - x^2\|_2. \end{aligned}$$

Proof. Given a fixed $\boldsymbol{\xi}$, by definition of $\bar{\nabla}Q^\xi$, we can derive

$$\bar{\nabla}Q_{sa}^\xi(x^1, \pi^1) - \bar{\nabla}Q_{sa}^\xi(x^2, \pi^2) = \frac{1}{J} \sum_{j=1}^J \sum_{t=0}^{T-1} \gamma^t \left(\nabla r_{s_t^j a_t^j}(x^1) - \nabla r_{s_t^j a_t^j}(x^2) \right).$$

The lipschitzness of ∇r leads to the lipschitzness of $\bar{\nabla}Q^\xi$. The proof for $\bar{\nabla}V^\zeta$ follows analogously. \square

Lemma I.4 (Lipschitzness of the hyper-gradient estimator). *Under Assumptions 7.2 and 7.3, for arbitrary upper-level variables $x^1, x^2 \in \mathbb{R}^n$, policies π^1, π^2 , we have*

$$\|\bar{\nabla}\phi(\mathbf{D}, \boldsymbol{\xi}, \boldsymbol{\zeta}; x^1, \pi^1) - \bar{\nabla}\phi(\mathbf{D}, \boldsymbol{\xi}, \boldsymbol{\zeta}; x^2, \pi^2)\|_2 \leq L_S \|x^1 - x^2\|_2, \quad (85)$$

with $L_S := \left(L_{l1} + \frac{2HI}{\tau(1-\gamma)} (C_l L_r + C_{rx} L_l) \right)$.

Proof. By definition (80), it holds that

$$\begin{aligned} \bar{\nabla}\phi(\mathbf{D}, \boldsymbol{\xi}, \boldsymbol{\zeta}; x^1, \pi^1) - \bar{\nabla}\phi(\mathbf{D}, \boldsymbol{\xi}, \boldsymbol{\zeta}; x^2, \pi^2) &= \frac{1}{M} \sum_{m=1}^M (\nabla l(\mathbf{d}^m; x^1) - \nabla l(\mathbf{d}^m; x^2)) \\ &+ \frac{1}{\tau M} \sum_{m=1}^M \left[l(\mathbf{d}^m; x^1) \sum_i \sum_h \bar{\nabla} \left(Q_{s_h^{m,i} a_h^{m,i}}^{\xi_k} - V_{s_h^{m,i}}^{\zeta_k} \right) (x^1, \pi^1) - l(\mathbf{d}^m; x^2) \sum_i \sum_h \bar{\nabla} \left(Q_{s_h^{m,i} a_h^{m,i}}^{\xi_k} - V_{s_h^{m,i}}^{\zeta_k} \right) (x^2, \pi^2) \right]. \end{aligned}$$

The boundedness and lipschitzness of $l, \nabla l, \bar{\nabla}Q^\xi$ and $\bar{\nabla}V^\zeta$ leads to the conclusion. \square

Define $p_k := \mathbb{E}_{\mathbf{D}_k, \xi_k, \zeta_k} [\bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; x_k, \pi_k)] - \tilde{\nabla}\phi(x_k, \pi_k)$ and $u_k := h_k - \mathbb{E}_{\mathbf{D}_k, \xi_k, \zeta_k} [\bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; x_k, \pi_k)]$, with h_k obeying the update rule,

$$\begin{aligned} h_k &= \bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; x_k, \pi_k) + (1 - \mu_k)(h_{k-1} - \bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; x_{k-1}, \pi_{k-1})) \\ &= \mu_k \bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; x_k, \pi_k) + (1 - \mu_k)(h_{k-1} + \bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; x_k, \pi_k) - \bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; x_{k-1}, \pi_{k-1})). \end{aligned}$$

The subsequent analysis is based on the following results established in Proposition I.2,

$$\left\| \mathbb{E}_{\mathbf{D}_k, \xi_k, \zeta_k} [\bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; x_k, \pi_k)] - \tilde{\nabla}\phi(x_k, \pi_k) \right\|_2 \leq b_\phi, \quad (86)$$

$$\mathbb{E}_{\mathbf{D}_k, \xi_k, \zeta_k} \left[\left\| \bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; x_k, \pi_k) - \mathbb{E}_{\mathbf{D}_k, \xi_k, \zeta_k} [\bar{\nabla}\phi(\mathbf{D}_k, \xi_k, \zeta_k; x_k, \pi_k)] \right\|_2^2 \right] \leq \sigma_\phi^2, \quad (87)$$

where b_ϕ and σ_ϕ^2 can be sufficiently small by adjusting the sampling configurations M, J, T . All the notation $\|\cdot\|$ without subscripts denotes the 2-norm for simplicity.

The next two lemmas characterize the descent properties of $\phi(x_k)$ and $\|u_k\|$ recursively.

Lemma I.5 (Descent property of the hyper-objective). *Under Assumptions 7.2 and 7.3, the iterates $\{x_k\}_{k=1}^K$ generated by Algorithm 4 satisfy*

$$\mathbb{E}[\phi(x_{k+1})] \leq \mathbb{E} \left[\phi(x_k) - \frac{\beta_k}{2} \|\nabla\phi(x_k)\|^2 - \frac{\beta_k}{2} (1 - \beta_k L_\phi) \|h_k\|^2 + \beta_k \|u_k\|^2 + 2\beta_k L_\phi^2 \epsilon_k^2 + 2\beta_k \|p_k\|^2 \right]. \quad (88)$$

where L_ϕ and $L_{\tilde{\phi}}$ are specified in Proposition H.6 and Proposition H.7, respectively, and the expectations are with respect to the stochasticity of the algorithm.

Proof. Based on the L_ϕ -Lipschitz smoothness of $\phi(x)$ revealed by Proposition H.6, a stochastic gradient step results in the following estimate of the hyper-objective φ :

$$\begin{aligned} \phi(x_{k+1}) &\leq \phi(x_k) + \langle \nabla\phi(x_k), x_{k+1} - x_k \rangle + \frac{L_\phi}{2} \|x_{k+1} - x_k\|^2 \\ &= \phi(x_k) - \beta_k \langle \nabla\phi(x_k), h_k \rangle + \frac{\beta_k^2 L_\phi}{2} \|h_k\|^2 \\ &= \phi(x_k) - \frac{\beta_k}{2} \|\nabla\phi(x_k)\|^2 - \frac{\beta_k}{2} (1 - \beta_k L_\phi) \|h_k\|^2 + \frac{\beta_k}{2} \|\nabla\phi(x_k) - h_k\|^2, \end{aligned} \quad (89)$$

where the last equality comes from $\langle a, b \rangle = \frac{1}{2}(\|a\|^2 + \|b\|^2 - \|a - b\|^2)$. Taking into account the equation $h_k - \tilde{\nabla}\phi(x_k, \pi_k) - p_k = u_k$ and Proposition H.7, we obtain

$$\begin{aligned} \|h_k - \nabla\phi(x_k)\|^2 &= \|h_k - \tilde{\nabla}\phi(x_k, \pi_k) - p_k + \tilde{\nabla}\phi(x_k, \pi_k) - \nabla\phi(x_k) + p_k\|^2 \\ &\leq 2 \|h_k - \tilde{\nabla}\phi(x_k, \pi_k) - p_k\|^2 + 4 \|\tilde{\nabla}\phi(x_k, \pi_k) - \nabla\phi(x_k)\|^2 + 4 \|p_k\|^2 \\ &\leq 2 \|u_k\|^2 + 4 L_{\tilde{\phi}}^2 \|\pi_k - \pi_k^*\|^2 + 4 \|p_k\|^2. \end{aligned}$$

Incorporating this inequality into (89) and taking expectation with respect to the algorithm lead to the result (88). \square

Lemma I.6 (Descent property of the estimation error). *Under Assumptions 7.2 and 7.3, the iterates $\{x_k\}_{k=1}^K$ generated by Algorithm 4 satisfy*

$$\begin{aligned} \mathbb{E} [\|u_{k+1}\|^2] &\leq 2(1 - \mu_{k+1})^2 \mathbb{E} [\|u_k\|^2] + 4\mu_{k+1}^2 \sigma_\phi^2 \\ &\quad + 16(1 - \mu_{k+1})^2 \left((L_S + L_\phi)^2 \beta_k^2 \mathbb{E} [\|h_k\|^2] + 2L_{\tilde{\phi}}^2 (\epsilon_k + \epsilon_{k+1}) + \mathbb{E} [\|p_{k+1}\|^2] + \mathbb{E} [\|p_k\|^2] \right), \end{aligned} \quad (90)$$

where $L_\phi, L_{\tilde{\phi}}$, and L_S are specified in Proposition H.6, Proposition H.7, and Lemma I.4; and the expectations are with respect to the stochasticity of the algorithm.

Proof. By definitions of h_k, u_k and p_k , we have

$$\begin{aligned}
 u_{k+1} &= h_{k+1} - \tilde{\nabla}\phi(x_{k+1}, \pi_{k+1}) - p_{k+1} \\
 &= (1 - \mu_{k+1})(h_k - \mathbb{E}_{\mathbf{D}_k, \boldsymbol{\xi}_k, \boldsymbol{\zeta}_k} [\bar{\nabla}\phi(\mathbf{D}_k, \boldsymbol{\xi}_k, \boldsymbol{\zeta}_k; x_k, \pi_k)]) + (1 - \mu_{k+1})\mathbb{E}_{\mathbf{D}_k, \boldsymbol{\xi}_k, \boldsymbol{\zeta}_k} [\bar{\nabla}\phi(\mathbf{D}_k, \boldsymbol{\xi}_k, \boldsymbol{\zeta}_k; x_k, \pi_k)] \\
 &\quad - (1 - \mu_{k+1})\bar{\nabla}\phi(\mathbf{D}_{k+1}, \boldsymbol{\xi}_{k+1}, \boldsymbol{\zeta}_{k+1}; x_k, \pi_k) + \bar{\nabla}\phi(\mathbf{D}_{k+1}, \boldsymbol{\xi}_{k+1}, \boldsymbol{\zeta}_{k+1}; x_{k+1}, \pi_{k+1}) \\
 &\quad - \tilde{\nabla}\phi(x_{k+1}, \pi_{k+1}) - p_{k+1} \\
 &= (1 - \mu_{k+1})u_k + \mu_{k+1}\left(\bar{\nabla}\phi(\mathbf{D}_{k+1}, \boldsymbol{\xi}_{k+1}, \boldsymbol{\zeta}_{k+1}; x_{k+1}, \pi_{k+1}) - \tilde{\nabla}\phi(x_{k+1}, \pi_{k+1}) - p_{k+1}\right) \\
 &\quad + (1 - \mu_{k+1})\left[\left(\bar{\nabla}\phi(\mathbf{D}_{k+1}, \boldsymbol{\xi}_{k+1}, \boldsymbol{\zeta}_{k+1}; x_{k+1}, \pi_{k+1}) - \tilde{\nabla}\phi(x_{k+1}, \pi_{k+1}) - p_{k+1}\right)\right. \\
 &\quad \left.- \left(\bar{\nabla}\phi(\mathbf{D}_{k+1}, \boldsymbol{\xi}_{k+1}, \boldsymbol{\zeta}_{k+1}; x_k, \pi_k) - \tilde{\nabla}\phi(x_k, \pi_k) - p_k\right)\right].
 \end{aligned}$$

This equality yields the following estimation

$$\mathbb{E} [\|u_{k+1}\|^2] \leq 2(1 - \mu_{k+1})^2 \mathbb{E} [\|u_k\|^2] \quad (91)$$

$$\begin{aligned}
 &\quad + 4\mu_{k+1}^2 \mathbb{E} \left[\left\| \bar{\nabla}\phi(\mathbf{D}_{k+1}, \boldsymbol{\xi}_{k+1}, \boldsymbol{\zeta}_{k+1}; x_{k+1}, \pi_{k+1}) - \tilde{\nabla}\phi(x_{k+1}, \pi_{k+1}) - p_{k+1} \right\|^2 \right] \\
 &\quad + 4(1 - \mu_{k+1})^2 \mathbb{E} \left[\left\| \left(\bar{\nabla}\phi(\mathbf{D}_{k+1}, \boldsymbol{\xi}_{k+1}, \boldsymbol{\zeta}_{k+1}; x_{k+1}, \pi_{k+1}) - \tilde{\nabla}\phi(x_{k+1}, \pi_{k+1}) - p_{k+1} \right) \right. \right. \\
 &\quad \left. \left. - \left(\bar{\nabla}\phi(\mathbf{D}_{k+1}, \boldsymbol{\xi}_{k+1}, \boldsymbol{\zeta}_{k+1}; x_k, \pi_k) - \tilde{\nabla}\phi(x_k, \pi_k) - p_k \right) \right\|^2 \right]. \quad (92)
 \end{aligned}$$

Subsequently, notice that

$$\begin{aligned}
 &\left\| \left(\bar{\nabla}\phi(\mathbf{D}_{k+1}, \boldsymbol{\xi}_{k+1}, \boldsymbol{\zeta}_{k+1}; x_{k+1}, \pi_{k+1}) - \tilde{\nabla}\phi(x_{k+1}, \pi_{k+1}) - p_{k+1} \right) \right. \\
 &\quad \left. - \left(\bar{\nabla}\phi(\mathbf{D}_{k+1}, \boldsymbol{\xi}_{k+1}, \boldsymbol{\zeta}_{k+1}; x_k, \pi_k) - \tilde{\nabla}\phi(x_k, \pi_k) - p_k \right) \right\| \\
 &\leq \left\| \bar{\nabla}\phi(\mathbf{D}_{k+1}, \boldsymbol{\xi}_{k+1}, \boldsymbol{\zeta}_{k+1}; x_{k+1}, \pi_{k+1}) - \bar{\nabla}\phi(\mathbf{D}_{k+1}, \boldsymbol{\xi}_{k+1}, \boldsymbol{\zeta}_{k+1}; x_k, \pi_k) \right\| + \left\| \nabla\phi(x_{k+1}) - \tilde{\nabla}\phi(x_{k+1}, \pi_{k+1}) \right\| \\
 &\quad + \left\| \nabla\phi(x_k) - \tilde{\nabla}\phi(x_k, \pi_k) \right\| + \left\| \nabla\phi(x_{k+1}) - \nabla\phi(x_k) \right\| + \|p_{k+1}\| + \|p_k\| \\
 &\stackrel{(i)}{\leq} L_S \|x_{k+1} - x_k\| + L_{\tilde{\phi}} \|\pi_{k+1} - \pi_{k+1}^*\| + L_{\tilde{\phi}} \|\pi_k - \pi_k^*\| + L_\phi \|x_{k+1} - x_k\| + \|p_{k+1}\| + \|p_k\| \\
 &\leq (L_S + L_\phi) \|x_{k+1} - x_k\| + L_{\tilde{\phi}} (\sqrt{\epsilon_k} + \sqrt{\epsilon_{k+1}}) + \|p_{k+1}\| + \|p_k\|,
 \end{aligned} \quad (93)$$

where (i) comes from the Lipschitz properties of $\nabla\phi$ and $\tilde{\nabla}\phi$ revealed by Proposition H.6 and Proposition H.7, and Lipschitz property of $\bar{\nabla}\phi$ revealed by Lemma I.4. Incorporating (93) into (92) and recalling the variance σ_ϕ in (87), we complete the proof. \square

Now, assembling the descent properties of (88) and (90), we derive the convergence results.

Theorem I.7. *Under Assumptions 7.2 and 7.3 and given the maximum iteration number K , we can choose appropriate sampling configurations $M \sim \mathcal{O}(K^{4/3})$, $J \sim \mathcal{O}(1)$, $T \sim \mathcal{O}(\log K)$, and set the parameters as follows,*

$$\beta_k = \frac{1}{(n_\beta + k)^{1/3}}, \quad \mu_{k+1} = \max \left\{ \frac{15}{16}, 1 - \beta_k^2 \right\}, \quad \epsilon_k = \beta_k^2 \quad \text{for } k = 1, 2, \dots, K, \quad (94)$$

with the integer $n_\beta \geq 3$ and

$$\frac{L_\phi}{(n_\beta + 1)^{1/3}} + \frac{2(L_S + L_\phi)}{(n_\beta + 1)^{2/3}} \leq 1. \quad (95)$$

Then the iterates $\{x_k\}$ generated by Algorithm 4 satisfy

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} [\|\nabla\phi(x_k)\|^2] = \mathcal{O} \left(\frac{\log(K)}{K^{2/3}} \right) = \tilde{\mathcal{O}} \left(K^{-\frac{2}{3}} \right),$$

where the expectation is with respect to the stochasticity of the algorithm.

Proof. We consider the merit function $\mathcal{S}_k := \phi(x_k) + \frac{\|u_k\|^2}{\beta_{k-1}}$. Substituting the descent property (90) of $\mathbb{E}[\|u_{k+1}\|^2]$ leads to the following inequality.

$$\begin{aligned} \mathbb{E} \left[\frac{\|u_{k+1}\|^2}{\beta_k} - \frac{\|u_k\|^2}{\beta_{k-1}} \right] &\leq \left(\frac{2(1-\mu_{k+1})^2}{\beta_k} - \frac{1}{\beta_{k-1}} \right) \mathbb{E}[\|u_k\|^2] + \frac{4\mu_{k+1}^2}{\beta_k} \sigma_\phi^2 \\ &\quad + \frac{16(1-\mu_{k+1})^2}{\beta_k} \left((L_S + L_\phi)^2 \beta_k^2 \mathbb{E}[\|h_k\|^2] + 2L_\phi^2(\epsilon_k + \epsilon_{k+1}) + \mathbb{E}[\|p_{k+1}\|^2] + \mathbb{E}[\|p_k\|^2] \right). \end{aligned} \quad (96)$$

By the choice (94) of β_k and μ_k , we have for $k = 1, 2, \dots, K$,

$$\begin{aligned} \frac{2(1-\mu_{k+1})^2}{\beta_k} &\leq \frac{16(1-\mu_{k+1})^2}{\beta_k} \leq \frac{1-\mu_{k+1}}{\beta_k} \leq \beta_k, \\ \frac{2(1-\mu_{k+1})^2}{\beta_k} - \frac{1}{\beta_{k-1}} &\leq \beta_k - \frac{1}{\beta_{k-1}} = (k+n_\beta)^{-1/3} - (k-1+n_\beta)^{-1/3} \leq -(k+n_\beta)^{-1/3} = -\beta_k. \end{aligned}$$

Taking the two estimations into (96) shows

$$\begin{aligned} \mathbb{E} \left[\frac{\|u_{k+1}\|^2}{\beta_k} - \frac{\|u_k\|^2}{\beta_{k-1}} \right] &\leq -\beta_k \mathbb{E}[\|u_k\|^2] + \frac{4\mu_{k+1}^2}{\beta_k} \sigma_\phi^2 + (L_S + L_\phi)^2 \beta_k^3 \mathbb{E}[\|h_k\|^2] \\ &\quad + 2\beta_k L_\phi^2(\epsilon_k + \epsilon_{k+1}) + \beta_k \left(\mathbb{E}[\|p_{k+1}\|^2] + \mathbb{E}[\|p_k\|^2] \right). \end{aligned} \quad (97)$$

Next, we combine (97) with (88) to get the descent property of \mathcal{S}_k .

$$\begin{aligned} \mathbb{E}[\mathcal{S}_{k+1} - \mathcal{S}_k] &\leq -\frac{\beta_k}{2} \mathbb{E}[\|\nabla \phi(x_k)\|^2] - \frac{\beta_k}{2} \left(1 - \beta_k L_\phi - 2\beta_k^2 (L_S + L_\phi)^2 \right) \mathbb{E}[\|h_k\|^2] \\ &\quad + \frac{4\mu_{k+1}^2}{\beta_k} \sigma_\phi^2 + 2\beta_k L_\phi^2(2\epsilon_k + \epsilon_{k+1}) + \beta_k \left(\mathbb{E}[\|p_{k+1}\|^2] + 3\mathbb{E}[\|p_k\|^2] \right) \\ &\leq -\frac{\beta_k}{2} \mathbb{E}[\|\nabla \phi(x_k)\|^2] + \frac{4\mu_{k+1}^2}{\beta_k} \sigma_\phi^2 + 2\beta_k L_\phi^2(2\epsilon_k + \epsilon_{k+1}) + \beta_k \left(\mathbb{E}[\|p_{k+1}\|^2] + 3\mathbb{E}[\|p_k\|^2] \right), \end{aligned}$$

where the last inequality follows from the choice (95) of n_β . Telescoping index from $k = 1$ to $k = K$ and dividing it by K yield

$$\frac{1}{K} \mathbb{E}[\mathcal{S}_{K+1} - \mathcal{S}_1] \leq -\frac{1}{K} \sum_{k=1}^K \frac{\beta_k}{2} \mathbb{E}[\|\nabla \phi(x_k)\|^2] + \frac{4\sigma_\phi^2}{K} \sum_{k=1}^K \frac{1}{\beta_k} + \frac{6L_\phi^2}{K} \sum_{k=1}^{K+1} \beta_k \epsilon_k + \frac{4}{K} \sum_{k=1}^{K+1} \beta_k \|p_k\|^2. \quad (98)$$

Rearrange (98) and employ $\beta_K \leq \beta_k$,

$$\frac{\beta_K}{2K} \sum_{k=1}^K \mathbb{E}[\|\nabla \phi(x_k)\|^2] \leq \frac{1}{K} \mathbb{E}[\mathcal{S}_1 - \mathcal{S}_{K+1}] + \frac{4\sigma_\phi^2}{K} \sum_{k=1}^K \frac{1}{\beta_k} + \frac{6L_\phi^2}{K} \sum_{k=1}^{K+1} \beta_k \epsilon_k + \frac{4}{K} \sum_{k=1}^{K+1} \beta_k \|p_k\|^2.$$

Dividing $\beta_K/2$ on both sides implies that

$$\begin{aligned} \frac{1}{K} \sum_{k=1}^K \mathbb{E}[\|\nabla \phi(x_k)\|^2] &\leq \frac{2}{\beta_K K} \mathbb{E}[\mathcal{S}_1 - \mathcal{S}_{K+1}] + \frac{8\sigma_\phi^2}{\beta_K K} \sum_{k=1}^K \frac{1}{\beta_k} + \frac{12L_\phi^2}{\beta_K K} \sum_{k=1}^{K+1} \beta_k \epsilon_k + \frac{8}{\beta_K K} \sum_{k=1}^{K+1} \beta_k \|p_k\|^2 \\ &\leq \frac{2(\mathcal{S}_1 - \phi^*)}{\beta_K K} + \frac{8\sigma_\phi^2}{\beta_K K} \sum_{k=1}^K \frac{1}{\beta_k} + \frac{12L_\phi^2}{\beta_K K} \sum_{k=1}^{K+1} \beta_k \epsilon_k + \frac{8}{\beta_K K} \sum_{k=1}^{K+1} \beta_k \|p_k\|^2, \end{aligned} \quad (99)$$

where ϕ^* is the minimum of $\phi(x)$. By Proposition I.2, we can choose the appropriate sampling parameters $M \sim \mathcal{O}(K^{4/3})$, $T \sim \mathcal{O}(\log K)$ such that $\|p_k\|^2 \leq b_\phi^2 = K^{-2/3}$ and $\sigma_\phi^2 = K^{-4/3}$ in (86) and (87). With the parameters

specified in (94), we have $\sigma_\phi^2 \sum_{k=1}^{K+1} \frac{1}{\beta_k} = \mathcal{O}(1)$, $\sum_{k=1}^{K+1} \beta_k \epsilon_k = \sum_{k=1}^{K+1} \beta_k^3 = \mathcal{O}(\log K)$, $\sum_{k=1}^{K+1} \beta_k \|p_k\|^2 = \mathcal{O}(1)$, and therefore the conclusion,

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} [\|\nabla \phi(x_k)\|^2] = \mathcal{O}\left(\frac{\mathcal{S}_1 - \phi^*}{K^{2/3}}\right) + \mathcal{O}\left(\frac{1}{K^{2/3}}\right) + \mathcal{O}\left(\frac{\log K}{K^{2/3}}\right) + \mathcal{O}\left(\frac{1}{K^{2/3}}\right) = \tilde{\mathcal{O}}\left(K^{-\frac{2}{3}}\right).$$

□

The inequality (99) in the proof of Theorem I.7 shows $\frac{1}{K} \sum \mathbb{E}[\|\nabla \phi(x_k)\|_2^2] = \tilde{\mathcal{O}}(L_\phi^2 K^{-2/3})$, which means the outer iteration complexity is $\tilde{\mathcal{O}}(|\mathcal{A}|^{1.5} \epsilon^{-1.5})$ by substituting $L_\phi = \mathcal{O}(\sqrt{|\mathcal{A}|})$.

J DETAILS ON EXPERIMENTS

This section presents the details of the experiment implementation. We conduct the RLHF task on three Atari games from the Arcade Learning Environment (ALE) (Bellemare et al., 2013) and three Mujoco environments (Todorov et al., 2012) to empirically validate the performance of the model-free algorithm, SoBiRL. The reward provided by the original environments serves as the ground truth, and for each trajectory pair, preference is assigned to the trajectory with higher accumulated ground-truth reward. We compare SoBiRL with the bilevel algorithms DRLHF (Christiano et al., 2017), PBRL (Shen et al., 2024), HPGD Thoma et al. (2024), and a baseline algorithm SAC (Haarnoja et al., 2018a,b). All the bilevel solvers harness deep neural networks to predict rewards, while the baseline SAC receives ground-truth rewards for training. Additionally, we also test on a synthetic bilevel RL experiment to verify the convergence rate of the model-based algorithm, M-SoBiRL. The experiments are produced on a workstation that consists of two Intel® Xeon® Gold 6330 CPUs (total 2×28 cores), 512GB RAM, and one NVIDIA A800 (80GB memory) GPU. We have made the code available on <https://github.com/UCAS-YanYang/SoBiRL>.

J.1 Practical SoBiRL

SoBiRL adopts deep neural networks to parameterize the policy and the reward model. In this way, we supplement Algorithm 1 with details, resulting in the practical version, Algorithm 5. The SAC algorithm capable of automating the temperature parameter (Haarnoja et al., 2018b) is chosen as the lower-level solver for SoBiRL, i.e., for a fixed reward network, SAC is invoked with several timesteps to update the policy network, the Q-value network and the adaptive temperature parameter (line 2 of Algorithm 5). Here, to focus more on the implementation of SoBiRL, we stepsize the details of SAC, which fully follows the principles in (Haarnoja et al., 2018b).

Algorithm 5 Practical SoBiRL

Input: iteration threshold K , inner iteration timesteps N , step size β , buffer \mathcal{D} storing trajectory pairs, initial reward network parameterized by θ_1^r , initial policy network parameterized by θ_0^π , initial Q-value network function parameterized by θ_0^Q , initial temperature parameter τ_0

- 1: **for** $k = 1, \dots, K$ **do**
- 2: $\theta_k^\pi, \theta_k^Q, \tau_k = \text{SAC}(\theta_{k-1}^\pi, \theta_{k-1}^Q, \tau_{k-1}, N)$ ▷ lower-level policy update
- 3: Rollout the policy $\pi_k = \pi_{\theta_k^\pi}$ in the upper-level MDP to obtain trajectory pairs ▷ collect trajectories
- 4: Query preference and update the buffer \mathcal{D} with the labeled pairs
- 5: Compute the estimator $\nabla_p \phi(\theta_k^r, \pi_k, \tau_k)$ using \mathcal{D} ▷ upper-level reward update
- 6: Implement an inexact hyper-gradient descent step

$$\theta_{k+1}^r = \theta_k^r - \beta \nabla_p \phi(\theta_k^r, \pi_k, \tau_k)$$
- 7: **end for**

Output: $(\theta_{K+1}^r, \theta_{K+1}^\pi)$

Adapting from $\tilde{\nabla}\phi(x_k, \pi_k)$ in Section 5, a practical version of hyper-gradient estimator $\nabla_p\phi(x_k, \pi_k, \tau_k)$ is used,

$$\begin{aligned} \nabla_p\phi(x_k, \pi_k, \tau_k) &= \mathbb{E}_{d_i \sim \rho(d; \pi_k)} [\nabla l(d_1, d_2, \dots, d_I; x_k)] \\ &\quad + \tau_k^{-1} \mathbb{E}_{d_i \sim \rho(d; \pi_k)} \left[l(d_1, d_2, \dots, d_I; x_k) \left(\sum_i \sum_h \nabla \left(r_{s_h^i a_h^i}(x_k) - \mathbb{E}_{a' \sim \pi_k(\cdot | s_h^i)} [r_{s_h^i a'}(x_k)] \right) \right) \right]. \end{aligned} \quad (100)$$

The main difference between (100) and (14) lies in two aspects: 1) $\nabla_p\phi(x_k, \pi_k, \tau_k)$ accommodates the adaptive temperature parameter τ_k ; 2) the term $\nabla \left(r_{s_h^i a_h^i}(x_k) - \mathbb{E}_{a' \sim \pi_k(\cdot | s_h^i)} [r_{s_h^i a'}(x_k)] \right)$ is employed as an one-depth truncation of $\tilde{\nabla} \left(Q_{s_h^i a_h^i}(x_k, \pi_k) - V_{s_h^i}(x_k, \pi_k) \right)$ in (14), inspired by the expressions,

$$\begin{aligned} \tilde{\nabla} V_s(x_k, \pi_k) &= \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \nabla r_{s_t, a_t}(x_k) \mid s_0 = s, \pi_k, \mathcal{M}_\tau(x_k) \right], \\ \tilde{\nabla} Q_{sa}(x_k, \pi_k) &= \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \nabla r_{s_t, a_t}(x_k) \mid s_0 = s, a_0 = a, \pi_k, \mathcal{M}_\tau(x_k) \right]. \end{aligned}$$

J.2 Experiment Settings for RLHF

The results on the Atari games—BeamRider, Seaquest, and SpaceInvaders—are shown in Figures 1 and 2. The results on the Mujoco environments—HalfCheetah, Walker2d, and Hopper—are presented in Figure 3. The overall performance is summarized in Table 2.

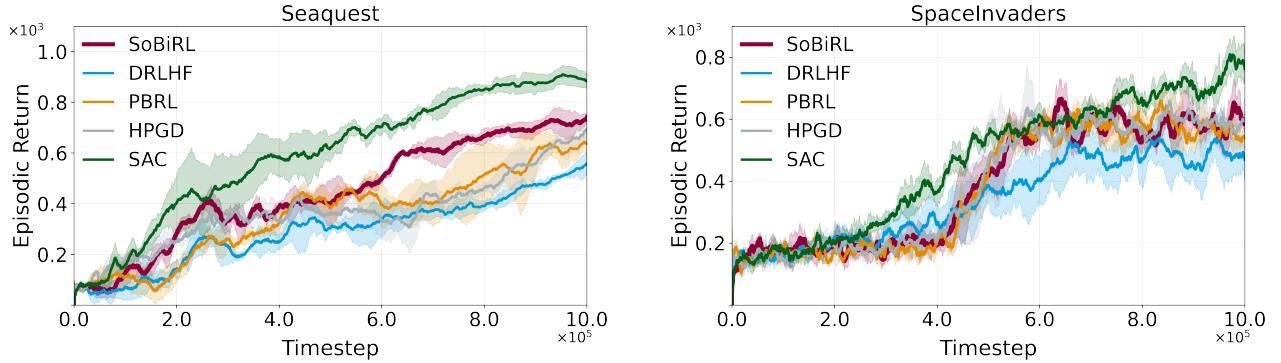


Figure 2: Comparison on the Atari games—Seaquest and SpaceInvaders—evaluated by the ground-truth reward. Each bilevel algorithm collects a total of 3000 trajectory pairs. The results are averaged over 5 seeds.

As mentioned above, we take SAC as the lower-level solver for the bilevel algorithms. We set the initial temperature parameter $\tau_0 = 1$, the inner iteration timesteps $N = 10^4$ for Atari games and $N = 5 \times 10^3$ for Mujoco simulations, and the batch size to 64. In addition, on Atari games, the actor network, the critic network, and the temperature parameter are updated by the Adam optimizer, with an initial learning rate 1×10^{-4} for BeamRider and SpaceInvaders, and 3×10^{-4} for Seaquest, linearly decaying to 0 after 8×10^6 time steps (although the runs were actually trained for only 4×10^6 timesteps). On all Mujoco simulations, the critic network and the temperature parameter share the same initial learning rate 1×10^{-3} , while that of the actor network is 3×10^{-4} . They are updated by Adam optimizers, linearly decaying the learning rates to 0 after 8×10^6 time steps (although the runs were actually trained for only 4×10^6 timesteps).

Bilevel algorithms employ deep neural networks to parameterize the reward model. The wrapped environment returns an $84 \times 84 \times 4$ tensor as the state information. Therefore, data in the shape of $84 \times 84 \times 4$ is fed into the reward network. It undergoes four convolutional layers with kernel sizes of 7×7 , 5×5 , 3×3 , 3×3 and stride values of 3, 2, 1, 1, respectively. Each convolutional layer holds 16 filters and incorporates leaky ReLU nonlinearities ($\alpha = 0.01$). Subsequently, the data passes through a fully connected layer of size 64 and is then transformed into a scalar. Batch normalization and dropout with a dropout rate of 0.5 are applied to all convolutional layers to mitigate overfitting. An AdamW (Loshchilov and Hutter, 2018) optimizer is adopted with the learning rate 3×10^{-4} , betas (0.9, 0.999), epsilon 10^{-8} and weight decay 10^{-2} .

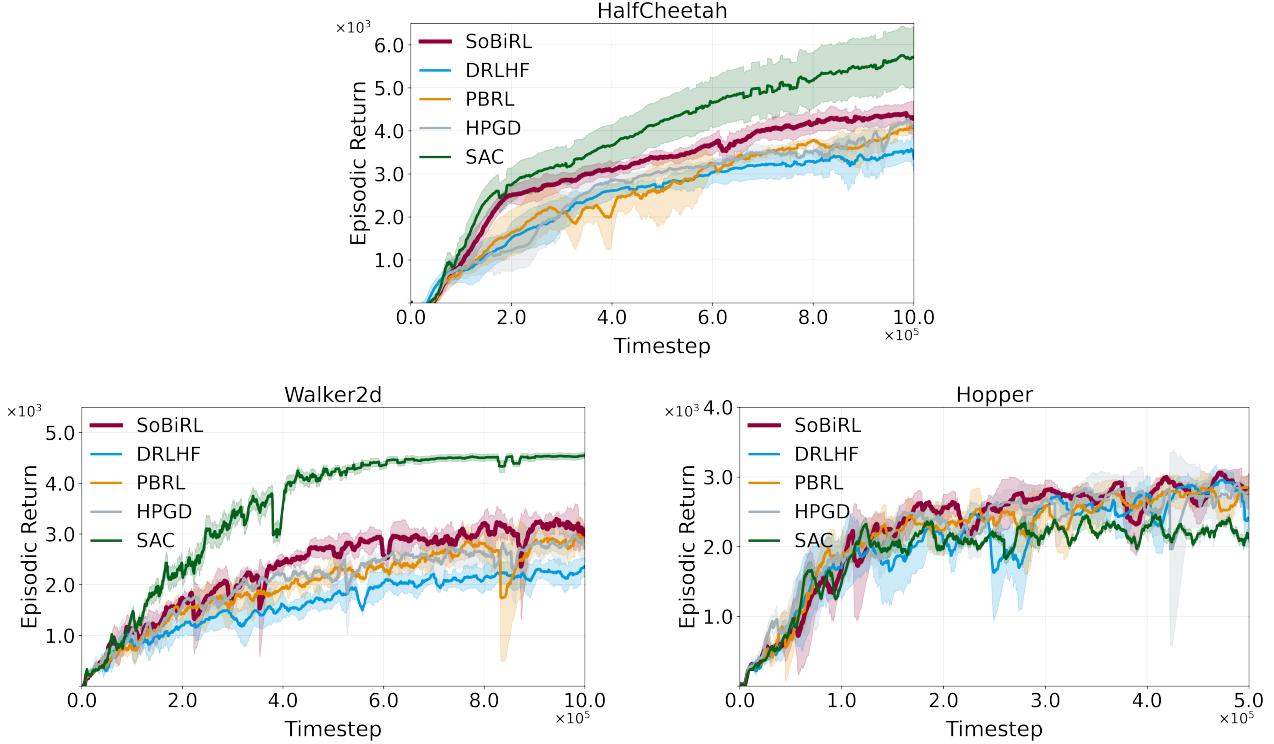


Figure 3: Comparison of algorithms on the Mujoco simulations—HalfCheetah, Walker2d, and Hopper—evaluated by the ground-truth reward. Each bilevel algorithm collects a total of 3000 trajectory pairs. The results are averaged over 5 seeds.

Trajectories of 25 timesteps are collected to construct the comparison pairs. Initially, we warm up the reward model by 500 epochs with 600 labeled pairs. In the following training, it collects 6 new pairs per reward learning epoch based on the current policy, until the buffer is filled with 3000 pairs. The batch size is set to 32.

Regarding the Atari games, wrappers are employed, which originate from (Mnih et al., 2015): initial 0 to 30 no-ops to inject stochasticity, max-pooling pixel values over the last two frames, an episodic life counter, four-frame skipping to accelerate sampling, four-frame stacking to help infer game dynamics, warping the image to size 84×84 and clipping rewards to $[-1, 1]$.

Specifically, after conducting initial tests, we observe that PBRL with the “value penalty” and the penalty parameter as 2 provide robust performance, for which it is adopted for comparison.

J.3 Experiment Settings for the Synthetic Bilevel RL Problem

To test M-SoBiRL, we build a synthetic bilevel RL problem in the form of (4) with the upper-level f as a quadratic function of (x, π) and the dimension of x as $n = 100$. The lower-level problem is adapted from RL problems used in (Zhan et al., 2023) and (Li et al., 2024b). Specifically, we take $|\mathcal{S}| = 10, |\mathcal{A}| = 5$ and $\gamma = 0.5, \tau = 1$. The transition probabilities are generated randomly, and for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, the reward r_{sa} is parameterized by a quadratic form of x with a random perturbation. The step sizes in Algorithm 2 are taken as $(\beta, \eta) = (0.008, 0.5)$. Results of the synthetic experiment are shown in Figure 4. The curves exhibit the benign convergence property of the proposed algorithm.

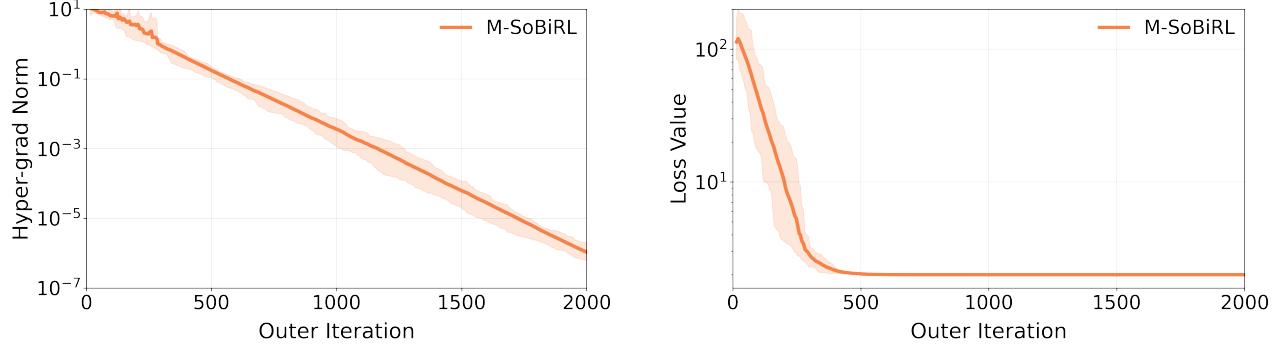


Figure 4: A synthetic bilevel RL problem to verify the model-based algorithm, M-SoBiRL. Metrics are the hyper-gradient norm $\|\nabla\phi(x)\|_2$ and the upper-level loss $f(x, \pi)$.

K APPLICATIONS OF BILEVEL REINFORCEMENT LEARNING

In this section, we provide a detailed explanation of how the formulation (4), with the upper-level function

$$f(x, \pi) = \mathbb{E}_{d_i \sim \rho(d; \pi)} [l(d_1, d_2, \dots, d_I; x)], \quad (101)$$

incorporates bilevel RL applications, using additional examples similar to those in Section 3.3. We also explain how the boundedness of l assumed in Section 7 can be verified individually for each case.

Reward Shaping From the perspective of bilevel RL, we can shape an auxiliary reward function $r(x)$ at the lower level for efficient agent training, while maintaining the original environment at the upper level to align with the initial task evaluation (Hu et al., 2020), which is established as follows,

$$\begin{aligned} \min_{x \in \mathbb{R}^n} -V_{\bar{\mathcal{M}}_{\bar{\tau}}}^{\pi^*(x)}(\bar{\rho}) &= -\mathbb{E}_{x \in \mathbb{R}^n} \left[Q_{\bar{\mathcal{M}}_{\bar{\tau}}}^{\pi^*(x)}(s, a) \mid s \sim \bar{\rho}, \pi^*(x) \right] \\ \text{s. t. } \pi^*(x) &= \arg \min_{\pi \in \Delta^{|\mathcal{A}|}} -V_{\mathcal{M}_{\tau}(x)}^{\pi}(\rho). \end{aligned}$$

Note that its upper-level function falls into the formulation (101) with $I = 1, H = 1, l = -Q_{\bar{\mathcal{M}}_{\bar{\tau}}}^{\pi}(s_0^1, a_0^1)$. Therefore, the boundedness and smoothness of l can be seen from Proposition H.5.

Reinforcement Learning from Human Feedback The target of RLHF is to learn the intrinsic reward function that incorporates expert knowledge, from simple labels only containing human preferences. It optimizes a policy under $r(x)$ at the lower level, and adjusts x to align the preference predicted by the reward model $r(x)$ with the true labels at the upper level.

$$\begin{aligned} \min_{x \in \mathbb{R}^n} & \mathbb{E}_{y, d_1, d_2 \sim \rho(d; \pi^*(x))} [l_h(d_1, d_2, y; x)] \\ \text{s. t. } \pi^*(x) &= \arg \min_{\pi \in \Delta^{|\mathcal{A}|}} -V_{\mathcal{M}_{\tau}(x)}^{\pi}(\rho), \end{aligned}$$

where each trajectory $d_i = \{(s_h^i, a_h^i)\}_{h=0}^{H-1}$ ($i = 1, 2$) is sampled from the distribution $\rho(d; \pi^*(x))$ generated by the policy $\pi^*(x)$ in the upper-level $\bar{\mathcal{M}}_{\bar{\tau}}$, i.e.,

$$P(d_i) = \bar{\rho}(s_0^i) [\Pi_{h=0}^{H-2} \pi^*(x)(a_h^i | s_h^i) \bar{P}(s_{h+1}^i | s_h^i, a_h^i)] \times \pi^*(x)(a_{H-1}^i | s_{H-1}^i),$$

and the preference label $y \in \{0, 1\}$, indicating preference for d_1 over d_2 , obeys human feedback distribution $y \sim D_{\text{human}}(y|d_1, d_2)$. Moreover, l_h is the binary cross-entropy loss, $l_h(d_1, d_2, y; x) = -y \log P(d_1 \succ d_2; x) - (1-y) \log P(d_2 \succ d_1; x)$, with the preference probability $P(d_1 \succ d_2; x)$ built by the Bradley–Terry model,

$$P(d_1 \succ d_2; x) = \frac{\exp\left(\sum_{h=0}^{H-1} r_{s_h^1, a_h^1}(x)\right)}{\exp\left(\sum_{h=0}^{H-1} r_{s_h^1, a_h^1}(x)\right) + \exp\left(\sum_{h=0}^{H-1} r_{s_h^2, a_h^2}(x)\right)}. \quad (102)$$

The upper-level function of RLHF is an instance of (101) with $I = 2$, finite H , and $l = \mathbb{E}_y[l_h(d_1, d_2, y; x)]$. Moreover, the boundedness of the reward $|r_{sa}(x)| \leq C_r$ will guarantee the boundedness of P , i.e., $\frac{1}{2}\exp(-2HC_r) \leq P(d_1 \succ d_2; x) \leq 1$, and thus the boundedness of l .

Contract Design This problem involves designing payment mechanisms that enable a principal to influence the agent's decision-making process (Wu et al., 2024). Through the state-contingent payment $x \in \mathbb{R}_+^{|\mathcal{S}| \times |\mathcal{S}|}$, the principal seeks to incentivize the agent to adopt policies that serve the principal's interests. Specifically, when the agent takes action a at state s , it costs $c(s, a)$ in the lower level while the principle only observes the transition $s \rightarrow s'$ and receives the reward $R(s, s')$. To encourage the transition, the principal offers a positive payment $x(s, s')$, which is thus added to the lower-level objective and subtracted from the upper-level objective.

$$\begin{aligned} \min_{x \in \mathbb{R}^n} & -\mathbb{E} \left[\sum_{t=0}^H R(s_t, s_{t+1}) - x(s_t, s_{t+1}) \mid s_0 \sim \rho, \pi^*(x) \right] \\ \text{s.t. } & \pi^*(x) = \arg \min_{\pi \in \Delta^{|\mathcal{A}|}} -\mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t (x(s_t, s_{t+1}) - c(s_t, a_t) + \tau h(\pi_{s_t})) \mid s_0 \sim \rho, \pi \right]. \end{aligned}$$

Note that its upper-level function is an instance of (101) with $I = 1$, finite H , and $l = \sum_{t=0}^H -R(s_t, s_{t+1}) + x(s_t, s_{t+1})$. It is reasonable that the reward R and the payment x are bounded in practice, and thus the function l is bounded.

Efficient Robot Navigation As formulated in (Chakraborty et al., 2024), we consider a maze-world environment of size $N \times N$, and aim to navigate a robot to get close to the destination (N, N) efficiently. To this end, the designer can introduce a modified goal $x \in \mathbb{R}^2$, which induces an associated reward $r_x \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$, and then use this goal to train the robot (in the lower level). In this manner, the upper-level objective includes both the received reward and the additional cost of moving the robot from x to (N, N) . The task is formulated by the following bilevel problem,

$$\begin{aligned} \min_{x \in \mathbb{R}^2} & \omega \|x - (N, N)\|^2 - \mathbb{E} \left[\sum_{t=0}^H r_x(s_t, a_t) \mid s_0 \sim \rho, \pi^*(x) \right] \\ \text{s.t. } & \pi^*(x) = \arg \min_{\pi \in \Delta^{|\mathcal{A}|}} -\mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t (r_x(s_t, a_t) + \tau h(\pi_{s_t})) \mid s_0 \sim \rho, \pi \right], \end{aligned}$$

where $\omega > 0$ is the weight balancing the deviation from the original destination (N, N) and the accumulative reward measured under the modified goal x . In this case, the upper-level function matches (101) with $I = 1$, finite H , and $l = -\sum_{t=0}^H (r_x(s_t, a_t) - \omega \|x - (N, N)\|^2 / H)$, which is bounded if the reward is bounded.

L EXTENSION TO ASYNCHRONOUS AND DISTRIBUTED BILEVEL RL SETTINGS

In Algorithm 1, the task within the k -th outer iteration is divided into two components: one is dedicated to approximately solving the lower-level problem such that $\|\pi_k^* - \pi_k\|_2^2 \leq \epsilon$, and the other is used to collect rollouts via π_k . We denote the sample complexity as c_{solver} and c_{roll} , respectively. Employing the asynchronous strategy, existing work (Shen et al., 2023) shows the actor-critic algorithm achieves a sample complexity c_{solver}/N per worker, where N is the number of workers. In a similar spirit, we can also take advantage of multiprocessing to collect rollouts amongst N workers asynchronously and thus enjoy c_{roll}/N complexity per worker. Finally, aggregate the rollouts from all the workers to estimate the hyper-gradient and update the upper-level variable. In

this manner, the error bound for the hyper-gradient estimation in our work still holds, and thus the convergence analysis can be generalized.

In the distributed setting (e.g., see ([Chen et al., 2022c](#))), agents are connected via a fully decentralized network and interact with a shared environment. Since the agent can communicate only with its neighbors, estimating the (global) hyper-gradient becomes challenging. Therefore, developing a tailored communication strategy and establishing a convergence analysis is an interesting and meaningful topic with broad applications in multi-agent RL problems. To this end, recent advances in decentralized bilevel optimization offer valuable insights. ([Kong et al., 2024](#); [He et al., 2024](#); [Zhu et al., 2024](#)).