Principled Penalty-based Methods for Bilevel Reinforcement Learning and RLHF

Han Shen [1] **Zhuoran Yang** [2] **Tianyi Chen** [1]

Abstract

Bilevel optimization has been recently applied to many machine learning tasks. However, their applications have been restricted to the supervised learning setting, where static objective functions with benign structures are considered. But bilevel problems such as incentive design, inverse reinforcement learning (RL), and RL from human feedback (RLHF) are often modeled as dynamic objective functions that go beyond the simple static objective structures, which pose significant challenges of using existing bilevel solutions. To tackle this new class of bilevel problems, we introduce the first principled algorithmic framework for solving bilevel RL problems through the lens of penalty formulation. We provide theoretical studies of the problem landscape and its penalty-based (policy) gradient algorithms. We demonstrate the effectiveness of our algorithms via simulations in the Stackelberg game and RLHF.

1. Introduction

Bilevel optimization (BLO) has emerged as an effective framework in machine learning. In a nutshell, BLO involves two coupled optimization problems in the upper and lower levels respectively, where they have different decision variables, denoted by x and y respectively. The lower-level problem is a constraint for the upper-level problem, e.g., in the upper level, we minimize a function $f(x, y)$ with the constraint that y is a solution to the lower-level problem determined by x, i.e., $y \in \mathcal{Y}^*(x)$. Here $\mathcal{Y}^*(x)$ is the solution set of the lower-level problem determined by x.

BLO enjoys a wide range of applications in machine learning, including hyper-parameter optimization (Franceschi et al., 2018), meta-learning (Finn et al., 2017; Rajeswaran et al., 2019), continue learning (Borsos et al., 2020), and adversarial learning (Jiang et al., 2021). Existing applications mostly concentrate on supervised learning setting, thus research on BLO has been predominantly confined to the static optimization setting (Franceschi et al., 2017), where in both the upper and lower-level problems, the objective functions are (strongly-)convex functions. However, this setting is insufficient to model more complex game-theoretic behaviors with sequential decision-making.

Reinforcement learning (RL) (Sutton & Barto, 2018) is a principled framework for sequential decision-making problems and has achieved tremendous empirical success in recent years (Silver et al., 2017; Ouyang et al., 2022). In this work, we study the BLO problem in the context of RL, where the lower-level problem is an RL problem and the upper-level problem can be either smooth optimization or RL. Specifically, in the lower-level problem, the follower solves a Markov decision process (MDP) determined by the leader's decision variable x, and returns a optimal policy of this MDP to the leader, known as the best response policy. The leader aims to maximize its own objective function, subject to the constraint that the follower always adopts the best response policy. This formulation of bilevel RL encompasses a range of applications such as Stackelberg Markov games (Stackelberg, 1952), reward learning (Hu et al., 2020), and RL from human feedback (RLHF) (Christiano et al., 2017). As an example, in RL from human feedback, the leader designs a reward r_x for the follower's MDP, with the goal that the resulting optimal policy yields the desired behavior of the leader.

Despite its various applications, the bilevel RL problem is difficult to solve. Broadly speaking, the main technical challenge lies in handling the constraint, i.e., the lower-level problem. The lower-level problem of bilevel RL extends from static smooth optimization to policy optimization in RL, and thus faces significant technical challenges. Such an extension loses a few optimization structures, such as strong convexity and uniform Polyak-Łojasiewicz (PL) condition, which are critical for existing BLO algorithms (Ghadimi & Wang, 2018; Shen & Chen, 2023). Specifically, there are *two mainstream approaches* for BLO: (a) implicit gradient

[1]Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, United States [2]Department of Statistics and Data Science, Yale University, United States. Correspondence to: Han Shen <shenh5@rpi.edu>, Tianyi Chen <chentianyi19@gmail.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

or iterative differentiation methods; and, (b) penalty-based methods. In (a), it is typically assumed that the lower-level objective function is strongly convex (Ji et al., 2021b; Chen et al., 2021), and thus its optimal solution $\mathcal{Y}^*(x)$ is unique. Then methods in (a) are essentially gradient-descent methods for the hyper objective $f(x, \mathcal{Y}^*(x))$, where the gradient of $\mathcal{Y}^*(x)$ can be computed using the implicit function theorem. However, in our bilevel RL case, the lower-level objective function is the discounted return in MDP, which is known to be non-convex (Agarwal et al., 2020). Thus, the hyper objective and its gradient are not well-defined. In (b), the bilevel problem is reformulated as a single-level problem by adding a penalty term of the lower-level suboptimality to the leader's objective function. The penalty reformulation approach has been studied in (Ye, 2012; Shen & Chen, 2023; Ye et al., 2022; Kwon et al., 2023) under the assumption that the lower-level objective function satisfies certain PL conditions. Unfortunately, when it comes to bilevel RL, the lower-level discounted return objective does not satisfy these conditions. To develop the penalty approach for bilevel RL problems, it is unclear (i) what is an appropriate penalty function; (ii) how is the solution to the reformulated problem related to the original bilevel problem; and, (iii) how to solve the reformulated problem. Therefore, directly extending applying BLO methods to bilevel RL is not straightforward, and new theories and algorithms tailored to the RL lower-level problem are needed, which are the subject of the paper.

1.1. Our contributions

To this end, we propose a novel algorithm that extends the idea of penalty-based BLO algorithm (Shen & Chen, 2023) to tackle the specific challenges of bilevel RL. Our approach includes the design of two tailored penalty functions: *value penalty and Bellman penalty*, which are crafted to capture the optimality condition of the lower-level RL problem.

In addition, leveraging the geometry of the policy optimization problem, we prove that an approximate solution to our reformulated problem is also an effective solution to the original bilevel problem.

Furthermore, we establish the differentiability of the reformulated problem and we propose a first-order policy-gradient-based algorithm. To our best knowledge, we establish the first provably convergent first-order algorithm for bilevel RL. Lastly, we conduct experiments on example applications covered by our framework, including the Stackelberg game and RL from human feedback tasks.

1.2. Related works

Bilevel optimization. The BLO problem can be dated back to (Stackelberg, 1952). The gradient-based BLO methods have gained growing popularity in the machine learning area; see, e.g., (Sabach & Shtern, 2017; Franceschi et al., 2018; Liu et al., 2020). A prominent branch of gradient-based BLO is based on the implicit gradient (IG) theorem. The IG based methods have been widely studied under a strongly-convex lower-level function, see, e.g., (Pedregosa, 2016; Ghadimi & Wang, 2018; Hong et al., 2023; Ji et al., 2021a; Chen et al., 2021; Khanduri et al., 2021; Shen & Chen, 2022; Li et al., 2022; Sow et al., 2022; Xiao et al., 2023b; Giovannelli et al., 2022; Chen et al., 2023). The iterative differentiation (ITD) methods, which can be viewed as an iterative relaxation of the IG methods, have been studied in, e.g., (Maclaurin et al., 2015; Franceschi et al., 2017; Nichol et al., 2018; Shaban et al., 2019; Liu et al., 2021b; 2022; Bolte et al., 2022; Grazzi et al., 2020; Ji et al., 2022; Shen & Chen, 2022). However, in our case the lower-level objective is the discounted return which is known to be non-convex (Agarwal et al., 2020). Thus it is difficult to apply the fore-mentioned methods here.

The penalty relaxation of the BLO problem, which can be dated back to (Clarke, 1983; Luo et al., 1996), has gained interest from researchers recently (see, e.g., (Shen & Chen, 2023; Ye et al., 2022; Lu & Mei, 2023; Kwon et al., 2023; Xiao et al., 2023a; Lu, 2024)). Theoretical results for this branch of work are established under certain lower-level error bounds weaker than strong convexity, but unfortunately not satisfied in our case. See Table 1 for more detailed comparison between this work and the general penalty-based BLO.

Policy-based RL. The policy-based RL algorithms are generally based on the policy gradient theorem (Sutton et al., 2000). There has been a large body of literature studying the policy-based algorithms, including the Monte-Carlo sampling based policy gradient methods (Sutton et al., 2000; Baxter & Bartlett, 2001), the advantage actor-critic algorithm (Borkar & Konda, 1997; Mnih et al., 2016), proximal policy optimization (Schulman et al., 2017), and more generally the policy mirror descent methods (Lan, 2023; Zhan et al., 2023). The landscape of the RL objective and the (global) convergence of the policy gradient based algorithms have been extensively studied in, to list a few, (Agarwal et al., 2020; Zhang et al., 2019; Qiu et al., 2019; Bhandari & Russo, 2019; Mei et al., 2020; Wu et al., 2020; Zhang et al., 2021; Cen et al., 2022; Shen et al., 2023; Ding et al., 2024).

Applications of bilevel RL. Bilevel RL covers several applications including reward shaping (Hu et al., 2020; Zou et al., 2019), reinforcement learning from preference (Christiano et al., 2017; Xu et al., 2020; Pacchiano et al., 2021), Stackelberg game (Liu et al., 2021a; Zhong et al., 2021; Song et al., 2023), AI-economics with two-level deep RL (Zheng et al., 2022), social environment design (Zhang et al., 2024), etc. A concurrent work (Chakraborty et al., 2024) studies the policy alignment problem, and introduces a corrected

Table 1: Comparison with general penalty-based BLO (e.g., (Shen & Chen, 2023; Kwon et al., 2023)). We compare this work with the smooth penalty case in previous works since the penalty functions are smooth in this work.

	Supervised penalty-based bilevel OPT	This work on penalty-based bilevel RL
Problem application	hyperparameter OPT, adversarial training, continue learning, etc.	Stackelberg Markov game, RL from preference, reward learning, etc
Penalty reformulation	Value penalty with assumed property	Value/Bellman penalty with proven property
Algorithm	Gradient directly accessible	Need to estimate gradients
Iteration complexity	$\tilde{\mathcal{O}}(\lambda\epsilon^{-1})$ with inner-loop GD	$\tilde{\mathcal{O}}(\lambda\epsilon^{-1})$ with inner-loop PMD

reward learning objective for RLHF that leads to strong performance gain. While PARL (Chakraborty et al., 2024) is based on the implicit gradient BLO method that requires the strong-convexity of the lower-level objective. On the other hand, PARL uses second-order derivatives of the RL objective, while our algorithm is fully first order. Finally, this work can be extended to the bilevel RL problem with multi-agent lower-level, thus including more applications, e.g., the incentive design (Yang et al., 2021).

2. Problem Formulations

In this section, we will first introduce the generic bilevel RL formulation. Then we will show several specific applications of the generic bilevel RL problem.

2.1. Bilevel reinforcement learning formulation

RL studies the problem where an agent aims to find a policy that maximizes its accumulated reward under the environment's dynamic. In such problem, the reward function and the dynamic are fixed given the agent's policy. While in the problem that we are about to study, the reward or the dynamic oftentimes depend on another decision variable, e.g., the reward is parameterized by a neural network in RLHF; or in Stackelberg game, both the reward and the dynamic are affected by the leader's policy.

Tailoring to this, we first define a so-called parameterized MDP. Given the parameter $x \in \mathbb{R}^{d_x}$, define a parameterized MDP as $\mathcal{M}_\tau(x) \coloneqq \{\mathcal{S}, \mathcal{A}, r_x, \mathcal{P}_x, \tau h\}$ where $\mathcal{S}$ is a finite state space; $\mathcal{A}$ is a finite action space; $r_x(s,a)$ is the parameterized reward given state-action pair $(s,a) \in \mathcal{S}\times\mathcal{A}$; $\mathcal{P}_x$ is a parameterized transition distribution that specifies $\mathcal{P}_x(s'|s,a)$–the probability of transiting to s' given (s,a); a policy π specifies $\pi(a|s)$ which is the probability of taking action a given state s; and τh is the regularization: $\tau \geq 0$ and $h = (h_s)_{s\in\mathcal{S}}$ where each $h_s : \Delta(\mathcal{A}) \mapsto \mathbb{R}_+$ is a strongly-convex regularization function given s. When $\tau = 0$, $\mathcal{M}_\tau(x)$ is an unregularized MDP.

Given a policy π, the value function of $\mathcal{M}_\tau(x)$ is defined as

$$V^\pi_{\mathcal{M}_\tau(x)}(s) \coloneqq \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty}\gamma^t\big(r_x(s_t,a_t) - \tau h_{s_t}(\pi(s_t))\big)\Big|s_0 = s\Big]$$

where $\gamma \in [0,1)$, $\pi(s) \coloneqq \pi(\cdot|s) \in \Delta(\mathcal{A})$ and the expectation is taken over the trajectory $(s_0, a_0 \sim \pi(s_0), s_1 \sim \mathcal{P}_x(\cdot|s_0,a_0), \dots)$. Given a state distribution ρ, we write $V^\pi_{\mathcal{M}_\tau(x)}(\rho) = \mathbb{E}_{s\sim\rho}[V^\pi_{\mathcal{M}_\tau(x)}(s)]$. Define the Q function as

$$Q^\pi_{\mathcal{M}_\tau(x)}(s,a) \coloneqq r_x(s,a) + \gamma\mathbb{E}_{s'\sim\mathcal{P}_x(\cdot|s,a)}\big[V^\pi_{\mathcal{M}_\tau(x)}(s')\big].$$

and $P^\pi_x(s_t = s|s_0)$ as the probability of reaching state s at time step t given initial state s_0 under a transition distribution $\mathcal{P}_x$ and a policy π. The probability $P^\pi_x(s_t = s|s_0, a_0)$ can be defined similarly.

Suppose the policy π is parameterized by $y \in \mathcal{Y} \subseteq \mathbb{R}^{d_y}$. We define the policy class as $\Pi \coloneqq \{\pi_y : y \in \mathcal{Y}\}$. We denote the optimal policy of $\mathcal{M}_\tau(x)$ as $\pi^*_y(x) \in \Pi$ satisfying $V^{\pi^*_y(x)}_{\mathcal{M}_\tau(x)}(s) \geq V^\pi_{\mathcal{M}_\tau(x)}(s)$ for any $\pi \in \Pi$ and s. With $f : \mathbb{R}^{d_x}\times\mathbb{R}^{d_y} \mapsto \mathbb{R}$, we are interested in solving

$$(2.1)\quad \mathcal{BM}: \min_{x,y} f(x,y), \ \text{s.t. } x \in \mathcal{X}, \quad y \in \mathcal{Y}^*(x) \coloneqq \operatorname*{argmin}_{y\in\mathcal{Y}} -V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho)$$

where $\mathcal{X} \subseteq \mathbb{R}^{d_x}$ and $\mathcal{Y} \subseteq \mathbb{R}^{d_y}$ are convex compact sets; and ρ is a given state distribution with $\rho(s) > 0$ on $\mathcal{S}$. The name 'bilevel' refers to the nested structure in the optimization problem: in the upper-level, a function $f(x,y)$ is minimized subject to the lower-level optimality constraint that π_y is the optimal policy for $\mathcal{M}_\tau(x)$.

2.2. Applications of bilevel reinforcement learning

Next we show several example applications that can be modeled by a bilevel RL problem.

Stackelberg Markov game. Consider a Markov game where at each time step, a leader and a follower observe the state and make actions simultaneously. Then according to the current state and actions, the leader and follower receive rewards and the game transits to the next state. Such a MDP can be defined as $\mathcal{M}^g_\tau \coloneqq \{\mathcal{S}, \mathcal{A}_l, \mathcal{A}_f, r_l, r_f, \mathcal{P}, \tau h_l, \tau h_f\}$ where $\mathcal{S}$ is the state space; $\mathcal{A}_l/\mathcal{A}_f$ is the leader's/follower's action space; $r_l(s,a_l,a_f)$ and $r_f(s,a_l,a_f)$ are respectively the leader's and the follower's reward given $(s,a_l,a_f) \in \mathcal{S}\times\mathcal{A}_l\times\mathcal{A}_f$; $\mathcal{P}(s'|s,a_l,a_f)$ is the probability of transiting to state s' given (s,a_l,a_f); the leader's/follower's policy π_x/π_y defines $\pi_x(a_l|s)/\pi_y(a_f|s)$–the probability of

choosing action a_l/a_f given state s; and $\tau h_l, \tau h_f$ are the regularization functions respectively for π_x and π_y.

Define the leader's/follower's value function as

$$V_\star^{\pi_x,\pi_y}(s) \coloneqq \mathbb{E}_{\pi_x,\pi_y}\Big[\sum_{t=0}^{\infty}\gamma^t\big(r_\star(s_t,a_{l,t},a_{f,t}) - \tau h_{\star,s_t}(\pi_\star(s_t))\big)\big|s_0=s\Big],\ \star = l \text{ or } f \quad (2.2)$$

where $\gamma \in [0,1)$, $\pi_\star(s) \coloneqq \pi_\star(\cdot|s) \in \Delta(\mathcal{A}_\star)$ and the expectation is taken over the trajectory $(s_0,\ a_{l,0}\sim\pi_x(s_0),\ a_{f,0}\sim\pi_y(s_0),\ s_{l,1}\sim\mathcal{P}(s_0,a_{l,0},a_{f,0}),\dots)$. Then the Q function can be defined as

$$Q_\star^{\pi_x,\pi_y}(s,a_l,a_f) \coloneqq r_\star(s,a_l,a_f) + \gamma\mathbb{E}_{s'\sim\mathcal{P}(s,a_l,a_f)}\big[V_\star^{\pi_x,\pi_y}(s')\big].$$

The follower's objective is to find a best-response policy to the leader's policy while the leader aims to find a best-response to the follower's best-response; that is

$$\max_{x,y} V_l^{\pi_x,\pi_y}(\rho), \text{s.t. } x\in\mathcal{X}, y\in\operatorname*{argmax}_{y\in\mathcal{Y}} V_f^{\pi_x,\pi_y}(\rho). \quad (2.3)$$

With the proof deferred to Appendix B.1, this problem can be viewed as a bilevel RL problem with $\mathcal{M}_\tau(x)$ in which $r_x(s,a_f) = \mathbb{E}_{a_l\sim\pi_x(s)}[r_l(s,a_l,a_f)]$ and $\mathcal{P}_x(\cdot|s,a_f) = \mathbb{E}_{a_l\sim\pi_x(s)}[\mathcal{P}(\cdot|s,a_l,a_f)]$.

Reinforcement learning from human feedback (RLHF). In the RLHF setting, the agent learns a task without knowing the true reward function. Instead, humans evaluate pairs of state-action segments, and for each pair they label the segment they prefer. The agent's goal is to learn the task well with limited amount of labeled pairs.

The original framework of deep RL from human feedback in (Christiano et al., 2017) (we call it DRLHF) consists of two possibly asynchronous learning process: reward learning from labeled pairs and RL from learnt rewards. In short, we maintain a buffer of labeled segment pairs $\{(d_0,l_0,d_1,l_1)_i\}_i$ where each segment $d = (s_t,a_t,\dots,s_{t+T},a_{t+T})$ is collected with the agent's policy π_y and l_0,l_1 is the label (e.g., $l_1=1,l_0=0$ indicates segment d_1 is preferred over d_0). DRLHF simultaneously learns a reward predictor r_x with the data and trains an RL agent using the learnt reward. This process has a hierarchy structure and can be reformulated as a bilevel RL problem:

$$\min_{x,y} -\mathbb{E}_{\pi_y}\big[l_0\log P(d_0\succ d_1|r_x) + l_1\log P(d_1\succ d_0|r_x)\big], \quad \text{s.t. } y\in\operatorname*{argmin}_y -V_{\mathcal{M}_\tau(x)}^{\pi_y}(\rho). \quad (2.4)$$

where $P(d_0\succ d_1|r_x) = \text{Sigmoid}(\sum_{s_t,a_t\in d_0} r_x(s_t,a_t) - \sum_{s_t,a_t\in d_1} r_x(s_t,a_t))$ is the probability of preferring d_0 over d_1 under reward r_x, given by the Bradley-Terry model.

Remark 1 (Connection with DPO (Rafailov et al., 2023)). The formulation in (2.4) becomes similar to DPO (Rafailov et al., 2023) in a special case. Specifically when $\gamma=0, T=0$, π_y is tabular and $h_s(\pi_y(s)) = D_{KL}(\pi_y(s)||\pi_{ref}(s))$ where π_{ref} is a given reference model, the lower level problem in (2.4) is solved if and only if the equation $r_x(s,a) = \tau\log\frac{\pi_y(a|s)}{\pi_{ref}(a|s)} + \tau\log Z_{r_x}(s)$ holds, where $Z_{r_x}(s)$ is some partition function (see, e.g., (Rafailov et al., 2023, eq. 5)). Plugging this equation back in the upper-level loss results in the DPO objective. The only difference is that the upper-level loss is on policy since the samples follow π_y, while the DPO loss depends on an off-policy dataset.

Reward shaping. In the RL tasks where the reward is difficult to learn from (e.g., the reward signal is sparse where most states give zero reward), we can reshape the reward to enable efficient policy learning while staying true to the original task. Given a task specified by $\mathcal{M}_\tau = \{\mathcal{S},\mathcal{A},r,\mathcal{P},\tau h\}$, the reward shaping problem (Hu et al., 2020) seeks to find a reshaped reward r_x such that the new MDP with r_x enables more efficient policy learning for the original task. We can define the new MDP as $\mathcal{M}_\tau(x) = \{\mathcal{S},\mathcal{A},r_x,\mathcal{P},\tau h\}$ and write the reward shaping problem as:

$$\min_{x,y} -V_{\mathcal{M}_\tau}^{\pi_y}(\rho), \text{s.t. } x\in\mathcal{X}, y\in\operatorname*{argmin}_{y\in\mathcal{Y}} -V_{\mathcal{M}_\tau(x)}^{\pi_y}(\rho) \quad (2.5)$$

which is a special case of bilevel RL.

3. Penalty Reformulation of Bilevel RL

A natural way to solve the bilevel RL problem $\mathcal{BM}$ is through reduction to a single-level problem, that is, to find a single-level problem that shares its local/global solutions with the original problem. Then by solving the single-level problem, we can recover the original solutions. In this section, we will perform single-level reformulation of $\mathcal{BM}$ through penalizing the upper-level objective with carefully chosen functions.

Specifically, we aim to find penalty functions $p(x,y)$ such that the solutions of the following problem recover the solutions of $\mathcal{BM}$:

$$\mathcal{BM}_{\lambda p}:\ \min_{x,y} F_\lambda(x,y) \coloneqq f(x,y) + \lambda p(x,y), \quad \text{s.t. } x\in\mathcal{X},\ y\in\mathcal{Y} \quad (3.1)$$

where λ is the penalty constant.

3.1. Value penalty and its landscape property

In $\mathcal{BM}$, the lower-level problem of finding the optimal policy π_y can be rewritten as its optimality condition: $-V_{\mathcal{M}_\tau(x)}^{\pi_y}(\rho) + \max_{y\in\mathcal{Y}} V_{\mathcal{M}_\tau(x)}^{\pi_y}(\rho) = 0$. Therefore, $\mathcal{BM}$ can be rewritten as

$$\min_{x\in\mathcal{X},y\in\mathcal{Y}} f(x,y), \text{s.t. } -V_{\mathcal{M}_\tau(x)}^{\pi_y}(\rho) + \max_{y\in\mathcal{Y}} V_{\mathcal{M}_\tau(x)}^{\pi_y}(\rho) = 0.$$

A natural penalty function that we call *value penalty* then measures the lower-level optimality gap:

$$p(x,y) = -V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho) + \max_{y\in\mathcal{Y}} V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho). \tag{3.2}$$

The value penalty specifies a penalized problem $\mathcal{BM}_{\lambda p}$ defined in (3.1). To capture the relation between solutions of $\mathcal{BM}_{\lambda p}$ and $\mathcal{BM}$, we have the following lemma.

Lemma 1 (Relation on solutions)**.** *Consider choosing p as the value penalty in (3.2). Assume there exists constant C such that $\max_{x\in\mathcal{X},y\in\mathcal{Y}} |f(x,y)| = \frac{C}{2}$. Given accuracy $\delta > 0$, choose $\lambda \geq C\delta^{-1}$. If (x_λ, y_λ) achieves ϵ-minimum of $\mathcal{BM}_{\lambda p}$, it achieves ϵ-minimum of the relaxed $\mathcal{BM}$:*

$$\begin{aligned}&\min_{x,y} f(x,y), \ \text{ s.t. } x\in\mathcal{X},\ y\in\mathcal{Y},\\ &\quad - V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho) + \max_{y\in\mathcal{Y}} V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho) \leq \epsilon_\lambda\end{aligned} \tag{3.3}$$

where $\epsilon_\lambda \leq \delta + \lambda^{-1}\epsilon$.

The proof is deferred to Appendix B.2. Perhaps one restriction of the above lemma is that it requires the boundedness of f on $\mathcal{X}\times\mathcal{Y}$. This assumption is usually mild in RL problems, e.g., it is guaranteed in Stackelberg game provided the reward functions are bounded.

Since $\mathcal{BM}_{\lambda p}$ is a non-convex problem, it is also of interest to connect the local solutions between $\mathcal{BM}_{\lambda p}$ and $\mathcal{BM}$. To achieve this, additional assumptions are required. Suppose we use direct policy parameterization: y is a vector with its (s,a) element $y_{s,a} = \pi_y(a|s)$, and thus $y = \pi_y$ directly. Then we can prove the following structural condition.

Lemma 2 (Gradient dominance)**.** *Given convex policy class Π and any $\tau \geq 0$, it holds for any $\pi\in\Pi$ that*

$$\max_{\pi'\in\Pi} \langle \nabla_\pi V^\pi_{\mathcal{M}_\tau(x)}(\rho),\, \pi' - \pi\rangle \geq \mu\big(\max_{\pi\in\Pi} V^\pi_{\mathcal{M}_\tau(x)}(\rho) - V^\pi_{\mathcal{M}_\tau(x)}(\rho)\big)$$

where $\mu = ((1-\gamma)\min_s \rho(s))^{-1}$.

See Appendix B.3 for a proof. A similar gradient dominance property was first proven in (Agarwal et al., 2020, Lemma 4.1) for the unregularized MDPs. The above lemma is a generalization of the result in (Agarwal et al., 2020) to regularized case. Under such structure of the lower-level problem, we arrive at the following lemma capturing the relation on local solutions.

Lemma 3 (Relation on local solutions)**.** *Consider using direct policy parameterization and choosing p as the value penalty in (3.2). Assume $f(x,\cdot)$ is L-Lipschitz-continuous on $\mathcal{Y}$. Given accuracy $\delta > 0$, choose $\lambda \geq LC_u\delta^{-1}$ where C_u is a constant specified in the proof. If (x_λ, y_λ) is a local solution of $\mathcal{BM}_{\lambda p}$, it is a local solution of the relaxed $\mathcal{BM}$ in (3.3) with an $\epsilon_\lambda \leq \delta$.*

The proof can be found in Appendix B.4. Lemmas 1 and 3 suggest we can recover the local/global solutions of the bilevel RL problem $\mathcal{BM}$ by locally/globally solving its penalty reformulation $\mathcal{BM}_{\lambda p}$ with the value penalty.

3.2. Bellman penalty and its property

Next we introduce the Bellman penalty that can be used as an alternative. To introduce this penalty function, we consider a tabular policy (direct parameterization) π_y, i.e. $\pi_y(\cdot|s) = y_s$ for all s and $y = (y_s)_{s\in\mathcal{S}} \in \mathcal{Y} = \Pi$. Then we can define the *Bellman penalty* as

$$p(x,y) = g(x,y) - v(x) \text{ where } v(x) := \min_{y\in\mathcal{Y}} g(x,y). \tag{3.4}$$

Here $g(x,y)$ is defined as

$$g(x,y) := \mathbb{E}_{s\sim\rho}[\langle y_s,\, q_s(x)\rangle + \tau h_s(y_s)] \tag{3.5}$$

where $q_s(x) \in \mathbb{R}^{|\mathcal{A}|}$ is the vector of optimal Q functions, which is defined as

$$q_s(x) = (q_{s,a}(x))_{a\in\mathcal{A}} \text{ where } q_{s,a}(x) := -\max_{\pi\in\Pi} Q^\pi_{\mathcal{M}_\tau(x)}(s,a).$$

It is immediate that $p(x,\cdot)$ is τ-strongly-convex uniformly for any $x\in\mathcal{X}$ by the 1-strong-convexity of h_s, and $p(x,y)\geq 0$ by definition. Moreover, we can show that the lower-level RL problem in $\mathcal{BM}$ is solved whenever $g(x,y) - v(x)$ is minimized in the following lemma.

Lemma 4. *Assume $\tau > 0$, then given any $x\in\mathcal{X}$, $\mathcal{M}_\tau(x)$ has a unique optimal policy $\pi^*_y(x)$. And we have $\arg\min_{y\in\mathcal{Y}} g(x,y) = \mathcal{Y}^*(x) = \{\pi^*_y(x)\}$. Therefore, $\mathcal{BM}$ can be rewritten as the following problem with $\epsilon = 0$:*

$$\begin{aligned}&\mathcal{BM}_\epsilon: \ \min_{x,y} f(x,y), \ \text{ s.t. } x\in\mathcal{X},\ y\in\mathcal{Y},\\ &g(x,y) - v(x) \leq \epsilon \text{ with } v(x) := \min_{y\in\mathcal{Y}} g(x,y).\end{aligned} \tag{3.6}$$

More generally for an $\epsilon > 0$, $\mathcal{BM}_\epsilon$ is an ϵ-approximate problem of $\mathcal{BM}$. A discussion on this and the proof of Lemma 4 are deferred to Appendix B.5. Based on Lemma 4, $g(x,y) - v(x)$ is a suitable lower-level optimality metric, thus is a natural penalty function candidate. We have the following result that proves the Bellman penalty is indeed a suitable penalty function.

Lemma 5 (Relation on solutions)**.** *Suppose choose the Bellman penalty in (3.4). Assume $f(x,\cdot)$ is L-Lipschitz-continuous on $\mathcal{Y}$. Given accuracy $\delta > 0$, choose $\lambda \geq L\sqrt{\tau^{-1}\delta^{-1}}$. If (x_λ, y_λ) is a local/global solution of $\mathcal{BM}_{\lambda p}$, then it is a local/global solution of $\mathcal{BM}_{\epsilon_\lambda}$ with $\epsilon_\lambda \leq \delta$.*

This lemma follows directly from the τ-strong-convexity of $g(x,\cdot)$ and (Shen & Chen, 2023, Proposition 3).

4. A Penalty-based Algorithm

In the previous sections, we have introduced two penalty functions $p(x,y)$ such that the original problem $\mathcal{BM}$ can be approximately solved via solving $\mathcal{BM}_{\lambda p}$. However, it is still unclear how $\mathcal{BM}_{\lambda p}$ can be solved. One challenge is the differentiability of the penalty function $p(x,y)$ in (3.1). In this section, we will first study the differentiability of $F_\lambda(x,y)$ and its specific gradient forms in each application. We will propose a penalty-based algorithm based on these results and further establish its convergence.

4.1. Differentiability of the value penalty

We first consider the value penalty

$$p(x,y) = -V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho) + \max_{y\in\mathcal{Y}} V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho).$$

For the differentiability in y, it follows $\nabla_y p(x,y) = -\nabla_y V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho)$ can be evaluated with the policy gradient theorem. The issue lies in the differentiability of $p(x,y)$ with respect to x, where $p(x,y)$ may not be differentiable in x due to the optimality function $\max_{y\in\mathcal{Y}} V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho)$. Fortunately, we will show that in the setting of RL, $p(\cdot,y)$ admits closed-form gradient under mild assumptions below.

Assumption 1. Assume (a) $\nabla_x V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho)$ is continuous in (x,y); and, (b) given any $x\in\mathcal{X}$ and $y,y'\in\mathcal{Y}^*(x)$, we have $\nabla_x V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho) = \nabla_x V^{\pi_{y'}}_{\mathcal{M}_\tau(x)}(\rho)$.

Assumption 1 (a) is mild in the applications, and can often be guaranteed by the a continuously differentiable reward function r_x. A sufficient condition of Assumption 1 (b) is the optimal policy of $\mathcal{M}_\tau(x)$ on Π is unique, e.g., when $\pi_y = \pi_{y'}$ for $y,y'\in\mathcal{Y}^*(x)$. As indicated by Lemma 4, the uniqueness is guaranteed when $\tau>0$.

Lemma 6 (Generic gradient form)**.** *Consider the value penalty p in (3.2). Suppose Assumption 1 holds. Then $p(x,y)$ is differentiable in x with the gradient*

$$\nabla_x p(x,y) = -\nabla_x V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho) + \nabla_x V^{\pi}_{\mathcal{M}_\tau(x)}(\rho)|_{\pi=\pi^*_y(x)}$$

where recall $\pi^*_y(x)$ is an optimal policy on policy class $\Pi=\{\pi_y : y\in\mathcal{Y}\}$ of $\mathcal{M}_\tau(x)$.

The proof can be found in Appendix C.1. Next, we can apply the generic result from Lemma 6 to specify the exact gradient formula in different bilevel RL applications discussed in Section 2.2.

Lemma 7 (Gradient form in the applications)**.** *Consider the value penalty p in (3.2). The gradient of the penalty function in specific applications are listed below.*

(a) RLHF/reward shaping: Assume r_x is continuously differentiable and Assumption 1 (b) holds. Then Lemma 6 holds and

$$\nabla_x p(x,y) = -\mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t \nabla r_x(s_t,a_t)\big|\rho,\pi_y\Big] + \mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t \nabla r_x(s_t,a_t)\big|\rho,\pi^*_y(x)\Big].$$

(b) Stackelberg game: Assume π_x is differentiable and Assumption 1 (b) holds. Then Lemma 6 holds and

$$\nabla_x p(x,y) = -\mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t \bar{Q}^{\pi_x,\pi_y}_{f,t}\nabla\log\pi_x(a_{l,t}|s_t)\big|s_0=s\Big] + \mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t \bar{Q}^{\pi_x,\pi^*_y(x)}_{f,t}\nabla\log\pi_x(a_{l,t}|s_t)\big|s_0=s\Big]$$

where $\bar{Q}^{\pi_x,\pi_y}_{f,t} := Q^{\pi_x,\pi_y}_f(s_t,a_{l,t},a_{f,t}) - \tau h_{f,s_t}(\pi_y(s_t))$. Recall in the Stackelberg setting, $\pi^*_y(x)$ is the optimal follower policy given π_x; and the expectation is taken over the trajectory generated by π_x,π_y(or $\pi^*_y(x)$), $\mathcal{P}$.

We defer the proof to Appendix C.2.

4.2. Differentiability of the Bellman penalty

For the Bellman penalty defined in (3.4), though it is straightforward to evaluate $\nabla_y p(x,y) = \nabla_y g(x,y)$, the differentiability of $p(x,y)$ in x is unclear. We next identify some sufficient conditions that allow convenient evaluation of $\nabla_x p(x,y)$.

Assumption 2. Assume $\tau>0$ and (a) given any (s,a), $\nabla_x Q^{\pi}_{\mathcal{M}_\tau(x)}(s,a)$ exists and is continuous in (x,π); and, (b) given $x\in\mathcal{X}$, for the MDP $\mathcal{M}_\tau(x)$, the Markov chain induced by any policy $\pi\in\Pi$ is irreducible[1].

Assumption 2 (a) is mild as it can be verified later in Lemma 8. Assumption 2 (b) is a regularity assumption on the MDP (Mitrophanov, 2005), and is often assumed in recent theoretical studies on policy gradient algorithms (see e.g., (Wu et al., 2020; Qiu et al., 2021)).

Lemma 8 (Generic gradient form)**.** *Consider the Bellman penalty in (3.4). Under Assumption 2, $p(x,y)$ is differentiable with $\nabla_x p(x,y) = \nabla_x g(x,y) - \nabla v(x)$ where*

$$\nabla_x g(x,y) = -\mathbb{E}_{s\sim\rho,a\sim\pi_y(s)}\big[\nabla_x Q^{\pi}_{\mathcal{M}_\tau(x)}(s,a)\big]\big|_{\pi=\pi^*_y(x)}$$

$$\nabla v(x) = -\mathbb{E}_{s\sim\rho,a\sim\pi(s)}\big[\nabla_x Q^{\pi}_{\mathcal{M}_\tau(x)}(s,a)\big]\big|_{\pi=\pi^*_y(x)}$$

The proof can be found in Appendix C.3. The above lemma provides the form of gradients for the $\mathcal{BM}$ problem. Next we show that Lemma 8 holds for the applications in Section 2.2 and then compute the closed-form of the gradients.

[1]The Markov chain is irreducible if for any state s and initial state-action pair s_0,a_0, there exists time step t such that $P^{\pi}_x(s_t=s|s_0,a_0)>0$, where $P^{\pi}_x(s_t=s|s_0,a_0)$ is the probability of reaching s at time step t in MDP $\mathcal{M}_\tau(x)$ with policy π.

Lemma 9 (Gradient form in the applications)**.** *Consider the Bellman penalty $p(x,y)$ in (3.4). The gradient form of the bilevel RL applications are listed below.*

(a) RLHF/reward shaping: Assume r_x is continuously differentiable and Assumption 2 (b) holds. Then Lemma 8 holds and

$$\nabla_x g(x,y) = -\mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t \nabla r_x(s_t,a_t)\big|a_0\sim\pi_y(s)\Big]$$

where the expectation is taken over $s_0\sim\rho, a_0\sim\pi_y(s_0)$ and the trajectory generated by $\pi_y^*(x)$ and $\mathcal{P}$, and $\nabla v(x) = \nabla_x g(x,y)|_{\pi_y=\pi_y^*(x)}$.

(b) Stackelberg game: Assume π_x is differentiable and Assumption 2 (b) holds. Then Lemma 8 holds and

$$\nabla_x g(x,y) = -\mathbb{E}\Big[Q_f^{\pi_x,\pi_y}(s_0,a_{l,0},a_{f,0})\nabla\log\pi_x(a_{l,0}|s_0) - \sum_{t=1}^{\infty}\gamma^t \bar{Q}_{f,t}^{\pi_x,\pi_y^*(x)}\nabla\log\pi_x(a_{l,t}|s_t)\Big]$$

where $\bar{Q}_{f,t}^{\pi_x,\pi_y} = Q_f^{\pi_x,\pi_y}(s_t,a_{l,t},a_{f,t}) - \tau h_{f,s_t}(\pi_y(s_t))$, $\pi_y^*(x)$ is the optimal follower policy given π_x; and the expectation is taken over all the randomness. Finally, we have $\nabla v(x) = \nabla_x g(x,y)|_{\pi_y=\pi_y^*(x)}$.

The proof is deferred to Appendix C.4.

4.3. A gradient-based algorithm and its convergence

In the previous subsections, we have addressed the challenges of evaluating $\nabla p(x,y)$, enabling the gradient-based methods to optimize $F_\lambda(x,y)$ in (3.1). However, computing $\nabla p(x_k,y_k)$ possibly requires an optimal policy $\pi_y^*(x_k)$ of the lower-level RL problem $\mathcal{M}_\tau(x_k)$. Given x_k, the lower-level RL problem can be solved with a wide range of algorithms, and we can use an approximately optimal policy parameter $\hat{\pi}_k \approx \pi_y^*(x_k)$ to compute the approximate penalty gradient $\hat{\nabla}p(x_k,y_k;\hat{\pi}_k) \approx \nabla p(x_k,y_k)$. The explicit formula of $\hat{\nabla}p(x_k,y_k;\hat{\pi}_k)$ can be straightforwardly obtained by replacing $\pi_y^*(x_k)$ with its approximate $\hat{\pi}_k$ in the formula of $\nabla p(x_k,y_k)$ in Lemmas 7 and 9. Therefore, we defer the formula to Appendix C.7 for ease of reading.

Given $\hat{\nabla}p(x_k,y_k;\hat{\pi}_k)$, we can compute the approximate gradient of F_λ as $\hat{\nabla}F_\lambda(x_k,y_k;\hat{\pi}_k) := \nabla f(x_k,y_k) + \lambda\hat{\nabla}p(x_k,y_k;\hat{\pi}_k)$ and update (with $\mathcal{Z}=\mathcal{X}\times\mathcal{Y}$)

$$(x_{k+1},y_{k+1}) = \mathrm{Proj}_{\mathcal{Z}}\Big[(x_k,y_k) - \alpha\hat{\nabla}F_\lambda(x_k,y_k;\hat{\pi}_k)\Big].$$

The optimization process is summarized in Algorithm 1.

Remark 2 (Comparison with the general penalty-based BLO algorithm.). The flow of the BLO-based algorithm in this work is similar to the general BLO algorithm in (Shen & Chen, 2023), the ingredients are significantly different; see our back-to-back comparison in Table 1. Specifically, the penalty gradient $\nabla p(x,y)$ is assumed to be directly accessible in the generic BLO algorithms. While in this work, we derive the close forms of $\nabla p(x,y)$ (Section 4), $\nabla p(x,y;\hat{\pi})$ (Appendix C.7) for our newly introduced penalty functions, and then use them in Algorithm 1.

Algorithm 1 PBRL: Penalty-based Bilevel RL Algorithm

1: Select either the value or Bellman penalty. Select $(x_1,y_1)\in\mathcal{Z}:=\mathcal{X}\times\mathcal{Y}$. Select step size α, penalty constant γ and iteration number K.
2: **for** $k=1$ **to** K **do**
3: Given RL problem $\mathcal{M}_\tau(x_k)$, compute an optimal policy estimator $\hat{\pi}_k\in\Pi$.
4: Compute the penalty's approximate gradient $\hat{\nabla}p(x_k,y_k;\hat{\pi}_k)\approx\nabla p(x_k,y_k)$.
5: Compute the inexact gradient of F_λ as $\hat{\nabla}F_\lambda(x_k,y_k;\hat{\pi}_k) = \nabla f(x_k,y_k) + \lambda\hat{\nabla}p(x_k,y_k;\hat{\pi}_k)$
6: $(x_{k+1},y_{k+1}) = \mathrm{Proj}_{\mathcal{Z}}\big[(x_k,y_k) - \alpha\hat{\nabla}F_\lambda(x_k,y_k;\hat{\pi}_k)\big]$
7: **end for**

We next study the convergence of PBRL. To bound the gradient error in Algorithm 1, we make the following assumption on the sub-optimality of the policy $\hat{\pi}_k$.

Assumption 3 (Oracle accuracy)**.** Given some accuracy ϵ_{orac} and step size α, assume the following inequality holds

$$\frac{1}{K}\sum_{k=1}^{K} 20\lambda^2\|\hat{\nabla}p(x_k,y_k;\hat{\pi}_k) - \nabla p(x_k,y_k)\|^2 \le \epsilon_{\mathrm{orac}} + \frac{1}{K}\sum_{k=1}^{K}\frac{1}{\alpha^2}\|(x_{k+1},y_{k+1}) - (x_k,y_k)\|^2. \quad (4.1)$$

This assumption only requires the running average of the error to be upper bounded, which is milder than requiring the error to be upper bounded for each iteration. A sufficient condition of the above assumption is $\|\hat{\pi}_k - \pi_y^*(x_k)\|^2 \le \epsilon_{\mathrm{orac}}$ with some constant c, which can be achieved by the policy mirror descent algorithm (see e.g., (Lan, 2023; Zhan et al., 2023)) with iteration complexity $\mathcal{O}(-\log(\epsilon_{\mathrm{orac}}/\lambda^2))$ (see a justification in Appendix C.7).

Furthermore, to guarantee worst-case convergence, the regularity condition that f and p are Lipschitz-smooth is required. We thereby identify a set of sufficient conditions for the value penalty or Bellman penalty to be smooth.

Assumption 4 (Smoothness assumption)**.** Assume $\forall(s,a)$, $h_s(\pi_y(s))$ is L_h-Lipschitz-smooth on $\mathcal{Y}$; and $Q_{\mathcal{M}_\tau(x)}^{\pi_y}(s,a)$, $V_{\mathcal{M}_\tau(x)}^{\pi_y}(s)$ are L_v-Lipschitz-smooth on $\mathcal{X}\times\mathcal{Y}$.

Assumption 4 is satisfied under a smooth r_x and a smooth policy (e.g., softmax policy (Mei et al., 2020)), or a direct policy parameterization paired with smooth regularization function h_s. See a discussion on this in Appendix C.5.

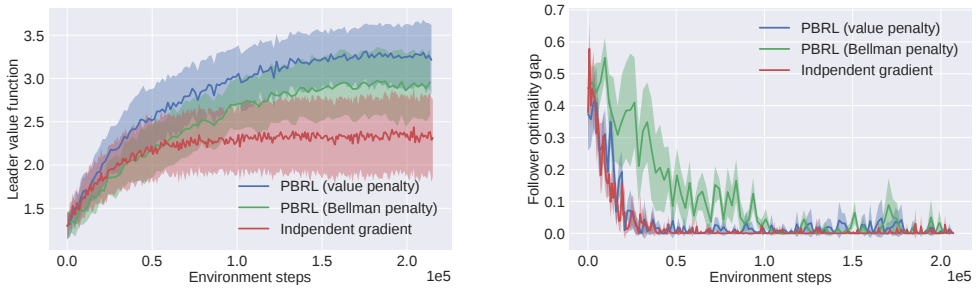

Figure 1. The result is generated by running the algorithms in 10 random Stackelberg MDPs. The environment step k is proportional to the number of samples used in training. The leader's value function is $V_l^{\pi_{x_k},\pi_{y_k}}(\rho)$, and the follower's optimality gap is given by $V_f^{\pi_{x_k},\pi_y^*(x_k)}(\rho) - V_f^{\pi_{x_k},\pi_{y_k}}(\rho)$. A zero optimality gap means the follower has found the best response to the leader.

Lemma 10 (Lipschitz smoothness of penalty functions)**.** *Under Assumptions 2 and 4, the value or Bellman penalty function $p(x,y)$ is L_p-Lipschitz-smooth on $\mathcal{X}\times\mathcal{Y}$ with constant L_p specified in the proof.*

We refer the reader to Appendix C.6 for a proof. Next, we make the final regularity assumption on f.

Assumption 5. Assume f is L_f-Lipschitz-smooth in (x,y).

The projected gradient is a commonly used metric in the convergence analysis of projected gradient type algorithms (Ghadimi et al., 2016). Define the projected gradient as

$$G_\lambda(x_k,y_k) := \alpha^{-1}\big((x_k,y_k) - (\bar{x}_{k+1},\bar{y}_{k+1})\big), \qquad (4.2)$$

where $(\bar{x}_{k+1},\bar{y}_{k+1}) := \mathrm{Proj}_{\mathcal{Z}}((x_k,y_k) - \alpha\nabla F_\lambda(x_k,y_k))$. Now we are ready to establish the convergence of PBRL.

Theorem 4.1 (Convergence of PBRL)**.** *Suppose Assumptions 2–5 hold. Choose step size $\alpha \le \frac{1}{L_f+\lambda L_p}$, then*

$$\frac{1}{K}\sum_{k=1}^{K}\|G_\lambda(x_k,y_k)\|^2 \le \frac{16\big(F_\lambda(x_1,y_1) - \inf_{\mathcal{Z}} f(x,y)\big)}{\alpha K} + \epsilon_{\mathrm{orac}}$$

See Appendix C.8 for the proof of above theorem. At each outer iteration k, let $\mathrm{com}(\epsilon_{\mathrm{orac}})$ be the oracle's iteration complexity. Then the above theorem suggests Algorithm 1 has an iteration complexity of $\mathcal{O}(\lambda\epsilon^{-1}\mathrm{com}(\epsilon_{\mathrm{orac}}))$. When choosing the oracle as policy mirror descent so that $\mathrm{com}(\epsilon_{\mathrm{orac}}) = \mathcal{O}(-\log(\epsilon_{\mathrm{orac}}/\lambda^2))$ (Lan, 2023; Zhan et al., 2023), Algorithm 1 has an iteration complexity of $\tilde{\mathcal{O}}(\lambda\epsilon^{-1})$.

5. Simulation

In this section, we test the empirical performance of PBRL.

5.1. Stackelberg Markov game

We first solve the following Stackelberg Markov game described in Section 2.2. We parameterize π_x and π_y with the softmax function. Here the transition distribution and rewards are randomly generated. It has a state space of size $|\mathcal{S}| = 100$, and the leader, and follower's action space are of size $|\mathcal{A}_l| = 5$, $|\mathcal{A}_f| = 5$ respectively. Each entry of the rewards $R_l, R_f \in \mathbb{R}^{100\times5\times5}$ is uniformly sampled between $[0,1]$ and values smaller than 0.7 are set to 0 to promote sparsity. Each entry of the transition matrix is sampled between $[0,1]$ and then is normalized to be a distribution.

Baseline. We implement PBRL with both value and Bellman penalty, and compare them with the independent policy gradient method (Daskalakis et al., 2020; Ding et al., 2022). In the independent gradient method, each player myopically maximizes its own value function, i.e., the leader maximizes $V_l^{\pi_x,\pi_y}(\rho)$ while the follower maximizes $V_f^{\pi_x,\pi_y}(\rho)$. At each step k, leader updates π_{x_k} with one-step gradient of $V_l^{\pi_x,\pi_{y_k}}(\rho)$ while the follower updates π_{y_k} with one-step gradient of $V_f^{\pi_{x_k},\pi_y}(\rho)$. We test all algorithms across 10 randomly generated MDPs.

We report the results in Figure 1. In the right figure, we can see the follower's optimality gap diminishes to zero, that is, the followers have found their optimal policies. In the mean time, the left figure reports the leaders' total rewards for the three methods. Overall, we find that both PBRL with value penalty and Bellman penalty outperform the independent gradient: it can be observed from Figure 1 (left) that PBRL can achieve a higher leader's return than the independent gradient, and the PBRL with value penalty reaches the highest value.

5.2. Deep reinforcement learning from human feedback

We test our algorithm in RLHF, following the experiment setting in (Christiano et al., 2017); see a description of the general RLHF setting in Section 2.2.

Environment and preference collection. We conduct our experiments in the Arcade Learning Environment (ALE) (Bellemare et al., 2013) through OpenAI gym. The ALE provides the game designer's reward that can be treated as the ground truth reward. For each pair of segments we collect, we assign preference to whichever has the highest ground truth reward. This allows us to benchmark our

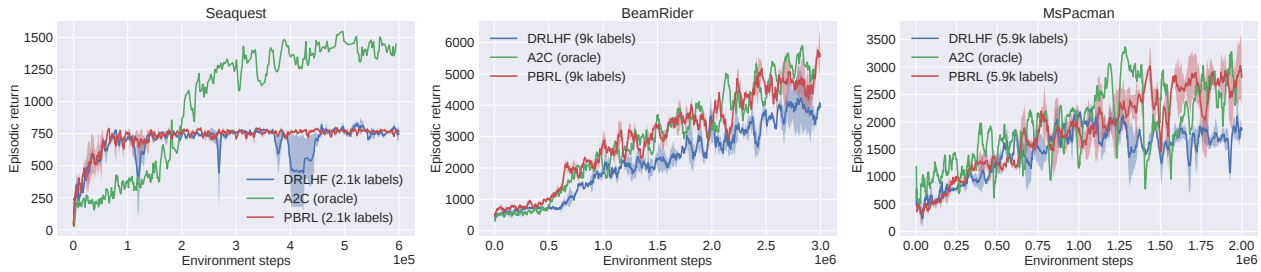

Figure 2. Performance on Atari games measured by true reward. The 'episode return' is the sum of true rewards in an episode. We average the episode return in 5 consecutive episodes. The 'environment steps' is the number of steps taken per worker in policy optimization. We compare performance of PBRL (ours) and DRLHF both with few labeled pairs, and A2C with true reward.

algorithm with DRLHF that also use this process.

Baseline. We compare PBRL with DRLHF (Christiano et al., 2017) and A2C (A3C (Mnih et al., 2016) but synchronous). We use the ground truth reward to train A2C agent, and treat A2C as an oracle algorithm which estimates a performance upperbound for other algorithms.

The results are reported in Figure 2. The first two games (Seaquest and BeamRider) are also reported in (Christiano et al., 2017). For Seaquest, the asymptotic performance of DRLHF and PBRL are similar, while DRLHF is more unstable in training. The unstability can also be made in the original paper of DRLHF. For BeamRider and MsPacman, we find out that PBRL has an advantage over DRLHF on the episode return. It can be observed that PBRL is able to achieve higher best-episode-return than DRLHF, and become comparable to the oracle algorithm.

6. Concluding Remarks

In this paper, we propose a penalty-based first-order algorithm for the bilevel RL problems. We provide results in three aspects: 1) we find penalty function with proper landscape properties such that the induced penalty reformulation admits solutions for the original bilevel RL problem; 2) to develop a gradient-based method, we study the differentiability of the penalty functions and find out their close form gradients; 3) based on the previous findings, we propose the convergent PBRL algorithm and evaluate it on the Stackelberg Markov game and the RLHF task.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgement

The work was supported by the National Science Foundation CAREER project 2047177 and the Cisco Research Award.

References

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. Optimality and approximation with policy gradient methods in markov decision processes. In *Proc. of Conference on Learning Theory*, 2020.

Baxter, J. and Bartlett, P. L. Infinite-horizon policy-gradient estimation. *journal of artificial intelligence research*, 15: 319–350, 2001.

Bellemare, M., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Bhandari, J. and Russo, D. Global optimality guarantees for policy gradient methods. *arXiv preprint arXiv:1906.01786*, 2019.

Bolte, J., Pauwels, E., and Vaiter, S. Automatic differentiation of nonsmooth iterative algorithms. In *Proc. of Advances in Neural Information Processing Systems*, 2022.

Borkar, V. and Konda, V. The actor-critic algorithm as multi-time-scale stochastic approximation. *Sadhana*, 22 (4):525–543, 1997.

Borsos, Z., Mutny, M., and Krause, A. Coresets via bilevel optimization for continual learning and streaming. In *Proc. of Advances in Neural Information Processing Systems*, 2020.

Cen, S., Cheng, C., Chen, Y., Wei, Y., and Chi, Y. Fast global convergence of natural policy gradient methods with entropy regularization. *Operations Research*, 70(4): 2563–2578, 2022.

Chakraborty, S., Bedi, A., Koppel, A., Manocha, D., Wang, H., Wang, M., and Huang, F. PARL: A unified framework for policy alignment in reinforcement learning. In *Proc. of International Conference on Learning Representations*, 2024.

Chen, T., Sun, Y., and Yin, W. Tighter analysis of alternating stochastic gradient method for stochastic nested problems. In *Proc. of Advances in Neural Information Processing Systems*, 2021.

Chen, X., Xiao, T., and Balasubramanian, K. Optimal algorithms for stochastic bilevel optimization under relaxed smoothness conditions. *arXiv preprint arXiv:2306.12067*, 2023.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Proc. of Advances in Neural Information Processing Systems*, 2017.

Clarke, F. *Optimization and non-smooth analysis*. Wiley-Interscience, 1983.

Clarke, F. H. Generalized gradients and applications. *Transactions of the American Mathematical Society*, 205:247–262, 1975.

Daskalakis, C., Foster, D. J., and Golowich, N. Independent policy gradient methods for competitive reinforcement learning. In *Proc. of Advances in Neural Information Processing Systems*, 2020.

Ding, D., Wei, C., Zhang, K., and Jovanovic, M. Independent policy gradient for large-scale markov potential games: Sharper rates, function approximation, and game-agnostic convergence. In *Proc. of International Conference on Machine Learning*, 2022.

Ding, D., Wei, C., Zhang, K., and Ribeiro, A. Last-iterate convergent policy gradient primal-dual methods for constrained mdps. In *Proc. of Advances in Neural Information Processing Systems*, 2024.

Dontchev, A. L. and Rockafellar, R. T. *Implicit functions and solution mappings: A view from variational analysis*, volume 616. Springer, 2009.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. of International Conference on Machine Learning*, 2017.

Franceschi, L., Donini, M., Frasconi, P., and Pontil, M. Forward and reverse gradient-based hyperparameter optimization. In *Proc. of International Conference on Machine Learning*, 2017.

Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., and Pontil, M. Bilevel programming for hyperparameter optimization and meta-learning. In *Proc. of International Conference on Machine Learning*, 2018.

Ghadimi, S. and Wang, M. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018.

Ghadimi, S., Lan, G., and Zhang, H. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1): 267–305, 2016.

Giovannelli, T., Kent, G., and Vicente, L. Inexact bilevel stochastic gradient methods for constrained and unconstrained lower-level problems. *arXiv preprint arXiv:2110.00604*, 2022.

Grazzi, R., Franceschi, L., Pontil, M., and Salzo, S. On the iteration complexity of hypergradient computation. In *Proc. of International Conference on Machine Learning*, pp. 3748–3758, 2020.

Hong, M., Wai, H.-T., Wang, Z., and Yang, Z. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM Journal on Optimization*, 33(1), 2023.

Hu, Y., Wang, W., Jia, H., Wang, Y., Chen, Y., Hao, J., Wu, F., and Fan, C. Learning to utilize shaping rewards: A new approach of reward shaping. In *Proc. of Advances in Neural Information Processing Systems*, 2020.

Ji, K., Yang, J., and Liang, Y. Provably faster algorithms for bilevel optimization and applications to meta-learning. In *Proc. of International Conference on Machine Learning*, 2021a.

Ji, K., Yang, J., and Liang, Y. Bilevel optimization: Convergence analysis and enhanced design. In *Proc. of International Conference on Machine Learning*, 2021b.

Ji, K., Liu, M., Liang, Y., and Ying, L. Will bilevel optimizers benefit from loops. In *Proc. of Advances in Neural Information Processing Systems*, 2022.

Jiang, H., Chen, Z., Shi, Y., Dai, B., and Zhao, T. Learning to defend by learning to attack. In *Proc. of International Conference on Artificial Intelligence and Statistics*, 2021.

Khanduri, P., Zeng, S., Hong, M., Wai, H.-T., Wang, Z., and Yang, Z. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. In *Proc. of Advances in Neural Information Processing Systems*, 2021.

Kwon, J., Kwon, D., Wright, S., and Nowak, R. On penalty methods for nonconvex bilevel optimization and first-order stochastic approximation. *arXiv preprint arXiv:2309.01753*, 2023.

Lan, G. Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes. *Mathematical programming*, 198(1), 2023.

Li, J., Gu, B., and Huang, H. A fully single loop algorithm for bilevel optimization without hessian inverse. In *Proc. of AAAI Conference on Artificial Intelligence*, 2022.

Liu, Q., Yu, T., Bai, Y., and Jin, C. A sharp analysis of model-based reinforcement learning with self-play. In *Proc. of International Conference on Machine Learning*, 2021a.

Liu, R., Mu, P., Yuan, X., Zeng, S., and Zhang, J. A generic first-order algorithmic framework for bi-level programming beyond lower-level singleton. In *Proc. of International Conference on Machine Learning*, 2020.

Liu, R., Liu, Y., Zeng, S., and Zhang, J. Towards gradient-based bilevel optimization with non-convex followers and beyond. In *Proc. of Advances in Neural Information Processing Systems*, 2021b.

Liu, R., Mu, P., Yuan, X., Zeng, S., and Zhang, J. A general descent aggregation framework for gradient-based bi-level optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):38–57, 2022.

Lu, S. Slm: A smoothed first-order lagrangian method for structured constrained nonconvex optimization. In *Proc. of Advances in Neural Information Processing Systems*, 2024.

Lu, Z. and Mei, S. First-order penalty methods for bilevel optimization. *arXiv preprint arXiv:2301.01716*, 2023.

Luo, Z., Pang, J., and Ralph, D. *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996.

Maclaurin, D., Duvenaud, D., and Adams, R. Gradient-based hyperparameter optimization through reversible learning. In *Proc. of International Conference on Machine Learning*, 2015.

Mei, J., Xiao, C., Szepesvari, C., and Schuurmans, D. On the global convergence rates of softmax policy gradient methods. In *Proc. of International Conference on Machine Learning*, 2020.

Mitrophanov, A. Y. Sensitivity and convergence of uniformly ergodic markov chains. *Journal of Applied Probability*, 42(4):1003–1014, 2005.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *Proc. of International Conference on Machine Learning*, 2016.

Nichol, A., Achiam, J., and Schulman, J. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. In *Proc. of Advances in Neural Information Processing Systems*, 2022.

Pacchiano, A., Saha, A., and Lee, J. Dueling rl: reinforcement learning with trajectory preferences. *arXiv preprint arXiv:2111.04850*, 2021.

Pedregosa, F. Hyperparameter optimization with approximate gradient. In *Proc. of International Conference on Machine Learning*, 2016.

Qiu, S., Yang, Z., Ye, J., and Wang, Z. On the finite-time convergence of actor-critic algorithm. In *Optimization Foundations for Reinforcement Learning Workshop at Advances in Neural Information Processing Systems*, 2019.

Qiu, S., Yang, Z., Ye, J., and Wang, Z. On finite-time convergence of actor-critic algorithm. *IEEE Journal on Selected Areas in Information Theory*, 2(2):652–664, 2021.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *Proc. of Advances in Neural Information Processing Systems*, 2023.

Rajeswaran, A., Finn, C., Kakade, S., and Levine, S. Meta-learning with implicit gradients. In *Proc. of Advances in Neural Information Processing Systems*, 2019.

Sabach, S. and Shtern, S. A first order method for solving convex bilevel optimization problems. *SIAM Journal on Optimization*, 27(2):640–660, 2017.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shaban, A., Cheng, C., Hatch, N., and Boots, B. Truncated back-propagation for bilevel optimization. In *Proc. of International Conference on Artificial Intelligence and Statistics*, 2019.

Shen, H. and Chen, T. A single-timescale analysis for stochastic approximation with multiple coupled sequences. In *Proc. of Advances in Neural Information Processing Systems*, 2022.

Shen, H. and Chen, T. On penalty-based bilevel gradient descent method. *arXiv preprint arXiv:2302.05185*, 2023.

Shen, H., Zhang, K., Hong, M., and Chen, T. Towards understanding asynchronous advantage actor-critic: Convergence and linear speedup. *IEEE Transactions on Signal Processing*, 2023.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Song, Z., Lee, J., and Yang, Z. Can we find nash equilibria at a linear rate in markov games? *arXiv preprint arXiv:2303.03095*, 2023.

Sow, D., Ji, K., and Liang, Y. On the convergence theory for hessian-free bilevel algorithms. In *Proc. of Advances in Neural Information Processing Systems*, 2022.

Stackelberg, H. *The Theory of Market Economy*. Oxford University Press, 1952.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction.* MIT Press, 2018.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Proc. of Advances in Neural Information Processing Systems*, 2000.

Wu, Y., Zhang, W., Xu, P., and Gu, Q. A finite time analysis of two time-scale actor critic methods. In *Proc. of Advances in Neural Information Processing Systems*, 2020.

Xiao, Q., Lu, S., and Chen, T. A generalized alternating method for bilevel learning under the polyak-łojasiewicz condition. In *Proc. of Advances in Neural Information Processing Systems*, 2023a.

Xiao, Q., Shen, H., Yin, W., and Chen, T. Alternating implicit projected sgd and its efficient variants for equality-constrained bilevel optimization. In *Proc. of International Conference on Artificial Intelligence and Statistics*, 2023b.

Xu, Y., Wang, R., Yang, L., Singh, A., and Dubrawski, A. Preference-based reinforcement learning with finite-time guarantees. In *Proc. of Advances in Neural Information Processing Systems*, 2020.

Yang, J., Wang, E., Trivedi, R., Zhao, T., and Zha, H. Adaptive incentive design with multi-agent meta-gradient reinforcement learning. *arXiv preprint arXiv:2112.10859*, 2021.

Ye, J. The exact penalty principle. *Nonlinear Analysis: Theory, Methods & Applications*, 75(3):1642–1654, 2012.

Ye, M., Liu, B., Wright, S., Stone, P., and Liu, Q. Bome! bilevel optimization made easy: A simple first-order approach. In *Proc. of Advances in Neural Information Processing Systems*, 2022.

Zhan, W., Cen, S., Huang, B., Chen, Y., Lee, J. D., and Chi, Y. Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence. *SIAM Journal on Optimization*, 33(2), 2023.

Zhang, E., Zhao, S., Wang, T., Hossain, S., Gasztowtt, H., Zheng, S., Parkes, D., Tambe, M., and Chen, Y. Social environment design. *arXiv preprint arXiv:2402.14090*, 2024.

Zhang, J., Kim, J., Donoghue, B., and Boyd, S. Sample efficient reinforcement learning with reinforce. In *Proc. of AAAI Conference on Artificial Intelligence*, 2021.

Zhang, K., Koppel, A., Zhu, H., and Başar, T. Global convergence of policy gradient methods to (almost) locally optimal policies. *SIAM Journal on Control and Optimization*, 58(6):3586–3612, 2019.

Zheng, S., Trott, A., Srinivasa, S., Parkes, D., and Socher, R. The ai economist: Taxation policy design via two-level deep multiagent reinforcement learning. *Science advances*, 8(18):eabk2607, 2022.

Zhong, H., Yang, Z., Wang, Z., and Jordan, M. I. Can reinforcement learning find stackelberg-nash equilibria in general-sum markov games with myopic followers? *arXiv preprint arXiv:2112.13521*, 2021.

Zou, H., Ren, T., Yan, D., Su, H., and Zhu, J. Reward shaping via meta-learning. *arXiv preprint arXiv:1901.09330*, 2019.

A. Preliminary results

Lemma 11 (Lipschitz continuous optimal policy)**.** *Given $x \in \mathcal{X}$, consider the optimal policies in a convex policy class Π of a parameterized MDP $\mathcal{M}_\tau(x)$. Suppose Assumption 2 holds, $\tau > 0$ and $\mathcal{X}$ is compact. Then the optimal policy $\pi_y^*(x)$ is unique and the following inequality hold:*

$$\|\pi_y^*(x) - \pi_y^*(x')\| \le \tau^{-1} C_J \|x - x'\|, \ \forall x, x' \in \mathcal{X} \tag{A.1}$$

where C_J is a constant specified in the proof.

Proof. By Lemma 4, the optimal policy of $\mathcal{M}_\tau(x)$ on a convex policy class Π is unique, given by

$$\pi_y^*(q(x)) = \operatorname*{argmin}_{\pi \in \Pi} J(q(x), \pi) \coloneqq \mathbb{E}_{s\sim\rho}[\langle \pi(s), \, q_s(x)\rangle + \tau h_s(\pi(s))] \tag{A.2}$$

where recall $q(x) = (q_s(x))_{s\in\mathcal{S}}$ with

$$q_s(x) = (-\max_{\pi\in\Pi} Q^\pi_{\mathcal{M}_\tau(x)}(s,a))_{a\in\mathcal{A}}. \tag{A.3}$$

We overload the notation π^* here with $\pi_y^*(q(x))$ which equals $\pi_y^*(x)$. In (A.2), since $\tau\mathbb{E}_{s\sim\rho}[h_s(\pi(s))]$ is τ-strongly convex at π on Π, $\pi_y^*(q(x))$ satisfies (A.2) if and only if it is a solution of the following parameterized variational inequality (VI)

$$\langle \nabla_\pi J(q(x), \pi), \, \pi - \pi' \rangle \le 0, \ \ \forall \pi' \in \Pi \tag{A.4}$$

where

$$\nabla_\pi J(q(x), \pi) = \Big(\rho(s) q_s(x) + \tau \rho(s) \nabla h_s(\pi(s))\Big)_{s\in\mathcal{S}}. \tag{A.5}$$

First, it can be checked that $\nabla_\pi J(q(x), \pi)$ is continuously differentiable at any $(q(x), \pi)$. Secondly, by the uniform strong convexity of $J(q(x), \cdot)$, given any $q(x)$, it holds that

$$(\pi - \pi')^\top \nabla_\pi^2 J\big(q(x), \pi_y^*(q(x))\big)(\pi - \pi') \ge \tau^{-1}\|\pi - \pi'\|^2. \tag{A.6}$$

Given these two properties of the VI, it then follows from (Dontchev & Rockafellar, 2009, Theorem 2F.7) that the solution mapping $\pi_y^*(q(x))$ is τ^{-1}-Lipschitz-continuous locally at any point $q(x)$. Thus $\pi_y^*(q(x))$ is τ^{-1}-Lipschitz-continuous in $q(x)$ globally, yielding

$$\begin{aligned} \|\pi_y^*(q(x)) - \pi_y^*(q(x'))\| &\le \tau^{-1}\|q(x) - q(x')\| \\ &\le \tau^{-1} \max_{x\in\mathcal{X}} \|\nabla q(x)\| \|x - x'\| \\ &= \tau^{-1} C_J \|x - x'\| \end{aligned} \tag{A.7}$$

where the second inequality follows from $q(x)$ is continuously differentiable by Lemma 8 and continuity of $\pi_y^*(x)$, and $C_J = \max_{x\in\mathcal{X}} \|\nabla q(x)\|$ is well-defined by compactness of $\mathcal{X}$. □

B. Proof in Section 2 and 3

B.1. Proof that Stackelberg Markov game is a bilevel RL problem

Lemma 12 (Stackelberg game cast as $\mathcal{BM}$)**.** *The Stackelberg MDP from the follower's viewpoint can be defined as a parametric MDP:*

$$\mathcal{M}_\tau(x) = \{\mathcal{S}, \mathcal{A}_f, r_x(s, a_f) = \mathbb{E}_{a_l\sim\pi_x(s)}[r_l(s, a_l, a_f)], \mathcal{P}_x(\cdot|s, a_f) = \mathbb{E}_{a_l\sim\pi_x(s)}[\mathcal{P}(\cdot|s, a_l, a_f)], \tau h_f\}.$$

Then $V_f^{\pi_x,\pi_y}(s) = V^{\pi_y}_{\mathcal{M}_\tau(x)}(s), \forall s$ and the original formulation of Stackelberg game in (2.3) can be rewritten as $\mathcal{BM}$:

$$\mathcal{SG}: \min_{x,y} -V_l^{\pi_x,\pi_y}(\rho), \ \ \text{s.t. } x \in \mathcal{X}, \ y \in \mathcal{Y}^*(x) = \operatorname*{argmin}_{y\in\mathcal{Y}} -V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho). \tag{B.1}$$

Proof. Recall that the follower's value function $V_f^{\pi_x,\pi_y}(s)$ under the leader's policy π_x and the follower's policy π_y is defined as

$$V_f^{\pi_x,\pi_y}(s) = \mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t\big(r_f(s_t,a_{l,t},a_{f,t}) - \tau h_{f,s_t}(\pi_y(s_t))\big)\big|s_0=s,\pi_x,\pi_y\Big] \tag{B.2}$$

where the leader's action $a_{l,t}\sim\pi_l(s_t)$, the follower's action $a_{f,t}\sim\pi_f(s_t)$, and the state transition follows $s_{t+1}\sim\mathcal{P}(\cdot|s_t,a_{l,t},a_{f,t})$.

It then follows from a expansion of the expectation in (B.2) that

$$\begin{aligned}
V_f^{\pi_x,\pi_y}(s) &= \mathbb{E}_{a_{l,0}\sim\pi_x(s_0),a_{f,0}\sim\pi_y(s_0)}\Big[r_f(s_0,a_{l,0},a_{f,0}) - \tau h_{f,s_0}(\pi_y(s_0))\big|s_0=s,\pi_x,\pi_y\Big]\\
&+\gamma\mathbb{E}_{\substack{a_{l,0}\sim\pi_x(s_0),a_{f,0}\sim\pi_y(s_0)\\ s_1\sim\mathcal{P}(s_0,a_{l,0},a_{f,0})\\ a_{l,1}\sim\pi_x(s_1),a_{f,1}\sim\pi_y(s_1)}}\Big[r_f(s_1,a_{l,1},a_{f,1}) - \tau h_{f,s_1}(\pi_y(s_1))\big|s_0=s,\pi_x,\pi_y\Big]+\dots\\
&= \mathbb{E}_{a_{f,0}\sim\pi_y(s_0)}\Big[r_x(s_0,a_{f,0}) - \tau h_{f,s_0}(\pi_y(s_0))\big|s_0=s,\pi_y\Big]\\
&+\gamma\mathbb{E}_{\substack{a_{f,0}\sim\pi_y(s_0)\\ s_1\sim\mathcal{P}_x(s_0,a_{f,0})\\ a_{f,1}\sim\pi_y(s_1)}}\Big[r_x(s_1,a_{f,1}) - \tau h_{f,s_1}(\pi_y(s_1))\big|s_0=s,\pi_y\Big]+\dots\\
&= V_{\mathcal{M}_\tau(x)}^{\pi_y}(s)
\end{aligned} \tag{B.3}$$

where recall $\mathcal{P}_x(s,a_f)=\mathbb{E}_{a_l\sim\pi_x(s)}[\mathcal{P}(\cdot|s,a_l,a_f)]$ and $r_x(s,a_f)=\mathbb{E}_{a_l\sim\pi_x(s)}[r_l(s,a_l,a_f)]$. Thus we have $V_f^{\pi_x,\pi_y}(s) = V_{\mathcal{M}_\tau(x)}^{\pi_y}(s),\forall s$. Therefore, the Stackelberg Markov game can be written as $\mathcal{BM}$. □

B.2. Proof of Lemma 1

Proof. Since (x_λ,y_λ) is an ϵ-minima of $\mathcal{BM}_{\lambda p}$, it holds for any $x\in\mathcal{X}$ and $y\in\mathcal{Y}$ that

$$f(x_\lambda,y_\lambda)+\lambda\big(-V_{\mathcal{M}_\tau(x_\lambda)}^{\pi_{y_\lambda}}(\rho)+\max_{y\in\mathcal{Y}}V_{\mathcal{M}_\tau(x_\lambda)}^{\pi_y}\big)\le f(x,y)+\lambda\big(-V_{\mathcal{M}_\tau(x)}^{\pi_y}(\rho)+\max_{y\in\mathcal{Y}}V_{\mathcal{M}_\tau(x)}^{\pi_y}\big)+\epsilon. \tag{B.4}$$

Choosing $x=x_\lambda$ and $y\in\mathcal{Y}(x_\lambda)$ in the above inequality and rearranging yields

$$\begin{aligned}
\max_{y\in\mathcal{Y}}V_{\mathcal{M}_\tau(x_\lambda)}^{\pi_y}(\rho) - V_{\mathcal{M}_\tau(x_\lambda)}^{\pi_{y_\lambda}}(\rho) &\le \frac{1}{\lambda}\big(f(x_\lambda,y_\lambda)-f(x_\lambda,y)+\epsilon\big)\\
&\le \frac{1}{\lambda}\big(C+\epsilon\big)\le\delta+\lambda^{-1}\epsilon.
\end{aligned} \tag{B.5}$$

Define $\epsilon_\lambda := \max_{y\in\mathcal{Y}}V_{\mathcal{M}_\tau(x_\lambda)}^{\pi_y}(\rho) - V_{\mathcal{M}_\tau(x_\lambda)}^{\pi_{y_\lambda}}(\rho)$ then $\epsilon_\lambda\le\delta+\lambda^{-1}\epsilon$. It follows from (B.4) that for any x,y feasible for (3.3) that

$$\begin{aligned}
f(x_\lambda,y_\lambda) &\le f(x,y)+\lambda\big(-V_{\mathcal{M}_\tau(x)}^{\pi_y}(\rho)+\max_{y\in\mathcal{Y}}V_{\mathcal{M}_\tau(x)}^{\pi_y}-\epsilon_\lambda\big)+\epsilon\\
&\le f(x,y)+\epsilon.
\end{aligned} \tag{B.6}$$

This completes the proof. □

B.3. Proof of Lemma 2

Proof. The following proof holds for any x and thus we omit x in the notations $\mathcal{M}_\tau(x)$, $\pi_y^*(x)$ and $\mathcal{P}_x$ in this proof. We first prove a policy gradient theorem for the regularized MDP. From the Bellman equation, we have

$$V_{\mathcal{M}_\tau}^{\pi}(s) = \sum_a\pi(a|s)Q_{\mathcal{M}_\tau}^{\pi}(s,a) - \tau h_s(\pi(s)) \tag{B.7}$$

Differentiating two sides of the equation with respect to π gives

$$\nabla V_{\mathcal{M}_\tau}^{\pi}(s) = \sum_a\nabla\pi(a|s)Q_{\mathcal{M}_\tau}^{\pi}(s,a)+\sum_a\pi(a|s)\nabla Q_{\mathcal{M}_\tau}^{\pi}(s,a) - \tau\nabla_\pi h_s(\pi(s)). \tag{B.8}$$

By the definition of Q function, we have $\nabla Q^\pi_{\mathcal{M}_\tau}(s,a) = \sum_{s'} \mathcal{P}(s'|s,a)\nabla V^\pi_{\mathcal{M}_\tau}(s')$. Substituting this inequality into (B.8) yields

$$\nabla V^\pi_{\mathcal{M}_\tau}(s) = \sum_a \nabla\pi(a|s) Q^\pi_{\mathcal{M}_\tau}(s,a) + \sum_{s'} P_\pi(s_1 = s'|s_0 = s)\nabla V^\pi_{\mathcal{M}_\tau}(s,a) - \tau\nabla_\pi h_s(\pi(s)) \tag{B.9}$$

where $P^\pi(s_1 = s'|s_0 = s)$ is the probability of $s_1 = s'$ given $s_0 = s$ under policy π. Note that the above inequality has a recursive structure, thus we can repeatedly applying it to itself and obtain

$$\nabla V^\pi_{\mathcal{M}_\tau}(s) = \frac{1}{1-\gamma}\mathbb{E}_{\bar{s}\sim d^\pi_s}\Big[\sum_a Q^\pi_{\mathcal{M}_\tau}(\bar{s},a)\nabla\pi(a|\bar{s})\Big] + \frac{\tau}{1-\gamma}\mathbb{E}_{\bar{s}\sim d^\pi_s}[-\nabla_\pi h_{\bar{s}}(\pi(\bar{s}))] \tag{B.10}$$

where $d^\pi_s(\bar{s}) := (1-\gamma)\sum_t \gamma^t P^\pi(s_t = \bar{s}|s_0 = s)$ is the discounted visitation distribution. Define $d^\pi_{\mathcal{M}_\tau}(\bar{s}) := \mathbb{E}_{s\sim\rho}[d^\pi_s(\bar{s})]$. Since $\nabla\pi(a|\bar{s}) = 1_{\bar{s},a}$ where $1_{\bar{s},a}$ is the indicator vector, we have the regularized policy gradient given by

$$\nabla_\pi V^\pi_{\mathcal{M}_\tau}(\rho) = \frac{1}{1-\gamma}\big[d^\pi_{\mathcal{M}_\tau}(s)\big(Q^\pi_{\mathcal{M}_\tau}(s,\cdot) - \tau\nabla h_s(\pi(s))\big)\big]_{s\in\mathcal{S}}. \tag{B.11}$$

Now we begin the prove the lemma. By the performance difference lemma (see e.g., (Lan, 2023, Lemma 2) and (Zhan et al., 2023, Lemma 5)), for any $\pi \in \Pi$, we have

$$\begin{aligned}
\max_{\pi\in\Pi} V^\pi_{\mathcal{M}_\tau}(\rho) - V^\pi_{\mathcal{M}_\tau}(\rho) &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi^*}_{\mathcal{M}_\tau}}\Big[\langle Q^\pi_{\mathcal{M}_\tau}(s,\cdot),\, \pi^*_y(s) - \pi(s)\rangle - \tau h_s(\pi^*_y(s)) + \tau h_s(\pi(s))\Big] \\
&\le \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi^*}_{\mathcal{M}_\tau}}\Big[\langle Q^\pi_{\mathcal{M}_\tau}(s,\cdot),\, \pi^*_y(s) - \pi(s)\rangle - \tau\langle\nabla h_s(\pi(s)),\, \pi^*_y(s) - \pi(s)\rangle\Big]
\end{aligned}$$

where the inequality follows from the convexity of h_s. Continuing from the inequality, it follows

$$\begin{aligned}
&\max_{\pi\in\Pi} V^\pi_{\mathcal{M}_\tau}(\rho) - V^\pi_{\mathcal{M}_\tau}(\rho) \\
&\le \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi^*}_{\mathcal{M}_\tau}}\Big[\max_{\pi'\in\Pi}\langle Q^\pi_{\mathcal{M}_\tau}(s,\cdot),\, \pi'(s) - \pi(s)\rangle - \tau\langle\nabla h_s(\pi(s)),\, \pi'(s) - \pi(s)\rangle\Big] \\
&= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^\pi_{\mathcal{M}_\tau}}\left[\frac{d^{\pi^*}_{\mathcal{M}_\tau}(s)}{d^\pi_{\mathcal{M}_\tau}(s)}\max_{\pi'\in\Pi}\Big(\langle Q^\pi_{\mathcal{M}_\tau}(s,\cdot),\, \pi'(s) - \pi(s)\rangle - \tau\langle\nabla h_s(\pi(s)),\, \pi'(s) - \pi(s)\rangle\Big)\right] \\
&\le \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^\pi_{\mathcal{M}_\tau}}\left[\left\|\frac{d^{\pi^*}_{\mathcal{M}_\tau}}{d^\pi_{\mathcal{M}_\tau}}\right\|_\infty \max_{\pi'\in\Pi}\Big(\langle Q^\pi_{\mathcal{M}_\tau}(s,\cdot),\, \pi'(s) - \pi(s)\rangle - \tau\langle\nabla h_s(\pi(s)),\, \pi'(s) - \pi(s)\rangle\Big)\right]
\end{aligned} \tag{B.12}$$

where the last inequality follows from $\frac{d^{\pi^*}_{\mathcal{M}_\tau}(s)}{d^\pi_{\mathcal{M}_\tau}(s)} \le \left\|\frac{d^{\pi^*}_{\mathcal{M}_\tau}}{d^\pi_{\mathcal{M}_\tau}}\right\|_\infty$ and

$$\begin{aligned}
&\max_{\pi'\in\Pi}\Big(\langle Q^\pi_{\mathcal{M}_\tau}(s,\cdot),\, \pi'(s) - \pi(s)\rangle - \tau\langle\nabla h_s(\pi(s)),\, \pi'(s) - \pi(s)\rangle\Big) \\
&\ge \langle Q^\pi_{\mathcal{M}_\tau}(s,\cdot),\, \pi(s) - \pi(s)\rangle - \tau\langle\nabla h_s(\pi(s)),\, \pi(s) - \pi(s)\rangle = 0.
\end{aligned} \tag{B.13}$$

Continuing from (B.12), we have

$$\begin{aligned}
&\max_{\pi\in\Pi} V^\pi_{\mathcal{M}_\tau}(\rho) - V^\pi_{\mathcal{M}_\tau}(\rho) \\
&\le \frac{1}{1-\gamma}\frac{1}{(1-\gamma)\min_s\rho(s)}\max_{\pi'\in\Pi}\mathbb{E}_{s\sim d^\pi_{\mathcal{M}_\tau}}\left[\Big(\langle Q^\pi_{\mathcal{M}_\tau}(s,\cdot),\, \pi'(s) - \pi(s)\rangle - \tau\langle\nabla h_s(\pi(s)),\, \pi'(s) - \pi(s)\rangle\Big)\right] \\
&= \frac{1}{(1-\gamma)\min_s\rho(s)}\max_{\pi'\in\Pi}\langle\nabla_\pi V^\pi_{\mathcal{M}_\tau}(\rho),\, \pi' - \pi\rangle
\end{aligned} \tag{B.14}$$

where the inequality follows from $(1-\gamma)\rho(s) \le d^\pi_{\mathcal{M}_\tau}(s) \le 1$ for any s and π, and the equality follows from (B.11). This proves the result. □

B.4. Proof of Lemma 3

Proof. Given x_λ, point y_λ satisfies the first-order stationary condition:

$$\langle \nabla_y f(x_\lambda, y_\lambda) + \lambda \nabla_y p(x_\lambda, y_\lambda),\, y_\lambda - y' \rangle \leq 0,\ \forall y' \in \mathcal{Y} \tag{B.15}$$

which leads to

$$\begin{aligned} \langle \nabla_y p(x_\lambda, y_\lambda),\, y_\lambda - y' \rangle &\leq -\frac{1}{\lambda} \langle \nabla_y f(x_\lambda, y_\lambda),\, y_\lambda - y' \rangle \\ &\leq \frac{L\|y_\lambda - y'\|}{\lambda} \leq \frac{LC_u}{\lambda},\ \forall y' \in \mathcal{Y} \end{aligned} \tag{B.16}$$

where $C_u \coloneqq \max_{y,y' \in \mathcal{Y}} \|y - y'\|$ which is well defined by compactness of $\mathcal{Y}$. For the LHS of the above inequality, we have the following inequality hold

$$\begin{aligned} \min_{y' \in \mathcal{Y}} \langle \nabla_y p(x_\lambda, y_\lambda),\, y_\lambda - y' \rangle &= \max_{y' \in \mathcal{Y}} \langle \nabla_y V^{\pi_{y_\lambda}}_{\mathcal{M}_\tau(x_\lambda)}(\rho),\, y' - y_\lambda \rangle \\ &\geq \frac{1}{(1-\gamma)\min_s \rho(s)} \big(\max_{y \in \mathcal{Y}} V^{\pi_y}_{\mathcal{M}_\tau(x_\lambda)}(\rho) - V^{\pi_{y_\lambda}}_{\mathcal{M}_\tau(x_\lambda)}(\rho) \big) \end{aligned} \tag{B.17}$$

where the last inequality follows from we are using direct policy parameterization $y = \pi$ and Lemma 2.

Substituting (B.17) into (B.16) yields

$$\max_{y \in \mathcal{Y}} V^{\pi_y}_{\mathcal{M}_\tau(x_\lambda)}(\rho) - V^{\pi_{y_\lambda}}_{\mathcal{M}_\tau(x_\lambda)}(\rho) \leq \frac{LC_u}{\lambda}. \tag{B.18}$$

Define $\epsilon_\lambda \coloneqq -V^{\pi_{y_\lambda}}_{\mathcal{M}_\tau(x_\lambda)}(\rho) + \max_{y \in \mathcal{Y}} V^{\pi_y}_{\mathcal{M}_\tau(x_\lambda)}(\rho)$ then $\epsilon_\lambda \leq \delta$ by choice of λ.

By local optimality of (x_λ, y_λ), it holds for any $x \in \mathcal{X}, y \in \mathcal{Y}$ and in the neighborhood of (x_λ, y_λ) that

$$f(x_\lambda, y_\lambda) + \lambda \big(- V^{\pi_{y_\lambda}}_{\mathcal{M}_\tau(x_\lambda)}(\rho) + \max_{y \in \mathcal{Y}} V^{\pi_y}_{\mathcal{M}_\tau(x_\lambda)} \big) \leq f(x, y) + \lambda \big(- V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho) + \max_{y \in \mathcal{Y}} V^{\pi_y}_{\mathcal{M}_\tau(x)} \big). \tag{B.19}$$

From the above inequality, it holds for any (x, y) feasible for the relaxed $\mathcal{BM}$ in (3.3) and in neighborhood of (x_λ, y_λ) that

$$\begin{aligned} f(x_\lambda, y_\lambda) &\leq f(x, y) + \lambda \big(- V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho) + \max_{y \in \mathcal{Y}} V^{\pi_y}_{\mathcal{M}_\tau(x)} - \epsilon_\lambda \big) \\ &\leq f(x, y) \end{aligned} \tag{B.20}$$

which proves the result. □

B.5. Proof of Lemma 4

Proof. Define

$$V^*_{\mathcal{M}_\tau(x)}(s) \coloneqq \max_{\pi \in \Pi} V^\pi_{\mathcal{M}_\tau(x)}(s), \quad Q^*_{\mathcal{M}_\tau(x)}(s, a) \coloneqq r(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}_x(s,a)}[V^*_{\mathcal{M}_\tau(x)}(s')].$$

Then it follows from the definition of the value function that for any s_0,

$$\begin{aligned} V^*_{\mathcal{M}_\tau(x)}(s_0) &= \max_{\pi \in \Pi} \mathbb{E}\Big[r_x(s_0, a_0) - \tau h_{s_0}(\pi(s_0)) + \sum_{t=1}^{\infty} \gamma^t \big(r_x(s_t, a_t) - \tau h_{s_t}(\pi(s_t)) \big) \big| s_0, \pi \Big] \\ &= \max_{\pi \in \Pi} \mathbb{E}\Big[r_x(s_0, a_0) - \tau h_{s_0}(\pi(s_0)) + \mathbb{E}\big[\sum_{t=1}^{\infty} \gamma^t \big(r_x(s_t, a_t) - \tau h_{s_t}(\pi(s_t)) \big) \big| s_0, a_0, \pi, \mathcal{P}_x \big] \big| s_0, \pi \Big] \\ &= \max_{\pi \in \Pi} \mathbb{E}_{a_0 \sim \pi(s_0)} \Big[r_x(s_0, a_0) - \tau h_{s_0}(\pi(s_0)) + \gamma \mathbb{E}_{s_1 \sim \mathcal{P}_x(s_0, a_0)} \big[V^\pi_{\mathcal{M}_\tau(x)}(s_1) \big] \Big] \\ &\leq \max_{\pi \in \Pi} \mathbb{E}_{a_0 \sim \pi(s_0)} \Big[r(s_0, a_0) - \tau h_{s_0}(\pi(s_0)) + \gamma \mathbb{E}_{s_1 \sim \mathcal{P}_x(s_0, a_0)} \big[V^*_{\mathcal{M}_\tau(x)}(s_1) \big] \Big] \end{aligned} \tag{B.21}$$

Given x, define a policy $\pi_y^* = (\pi_y^*(s))_{s\in\mathcal{S}} \in \Pi$ via

$$\pi_y^*(s_0) := \operatorname*{argmax}_{\pi(s_0)} \mathbb{E}_{a_0\sim\pi(s_0)}\Big[r(s_0,a_0) - \tau h_{s_0}(\pi(s_0)) + \gamma \mathbb{E}_{s_1\sim\mathcal{P}_x(s_0,a_0)}\big[V^*_{\mathcal{M}_\tau(x)}(s_1)\big]\Big], \forall s_0 \in \mathcal{S}$$

where the argmax is a singleton following from the τ-strong convexity of τh, and we sometimes treat the singleton set as its element for convenience. Given the definition of π_y^*, it then follows from (B.21) that

$$\begin{aligned} V^*_{\mathcal{M}_\tau(x)}(s_0) &\le \mathbb{E}_{a_0\sim\pi_y^*(s_0)}\Big[r(s_0,a_0) - \tau h_{s_0}(\pi(s_0)) + \gamma \mathbb{E}_{s_1\sim\mathcal{P}_x(s_0,a_0)}\big[V^*_{\mathcal{M}_\tau(x)}(s_1)\big]\Big] \\ &\le \mathbb{E}_{a_0\sim\pi_y^*(s_0)}\Big[r(s_0,a_0) - \tau h_{s_0}(\pi(s_0)) \\ &\quad + \gamma \mathbb{E}_{s_1\sim\mathcal{P}_x(s_0,a_0),a_1\sim\pi_y^*(s_1)}\big[r(s_1,a_1) - \tau h_{s_1}(\pi(s_1)) + \gamma \mathbb{E}_{s_2\sim\mathcal{P}_x(s_1,a_1)}[V^*_{\mathcal{M}_\tau(x)}(s_2)]\big]\Big] \end{aligned} \tag{B.22}$$

where the last inequality is a result of applying (B.21) twice. Continuing to recursively apply (B.21) and then using the definition of $V^\pi_{\mathcal{M}_\tau(x)}$ yield

$$V^*_{\mathcal{M}_\tau(x)}(s_0) \le V^{\pi_y^*}_{\mathcal{M}_\tau(x)}(s_0), \ \ \forall s_0 \in \mathcal{S} \tag{B.23}$$

which proves π_y^* is the optimal policy for $\mathcal{M}_\tau(x)$. In addition, we have

$$\begin{aligned} \pi_y^*(s_0) &= \operatorname*{argmax}_{\pi(s_0)} \mathbb{E}_{a_0\sim\pi(s_0)}\Big[r(s_0,a_0) - \tau h_{s_0}(\pi(s_0)) + \gamma \mathbb{E}_{s_1\sim\mathcal{P}_x(s_0,a_0)}\big[V^{\pi_y^*}_{\mathcal{M}_\tau(x)}(s_1)\big]\Big] \\ &= \operatorname*{argmax}_{\pi(s_0)} \mathbb{E}_{a_0\sim\pi(s_0)}\Big[Q^{\pi_y^*}_{\mathcal{M}_\tau(x)}(s_0,a_0) - \tau h_{s_0}(\pi(s_0))\Big], \ \ \forall s_0. \end{aligned} \tag{B.24}$$

Then we have $\pi_y^* = \arg\min_{y\in\Pi} g(x,y)$ and thus $\arg\min_{y\in\Pi} g(x,y) \in \mathcal{Y}^*(x)$. To further prove $\arg\min_{y\in\Delta(\mathcal{A})^{|\mathcal{S}|}} g(x,y) = \mathcal{Y}^*(x)$, it then suffices to prove any other policy $\pi \in \Pi$ different from π_y^* is not optimal. Let s_0' be the state such that $\pi_y^*(s_0') \ne \pi(s_0')$. We have

$$\begin{aligned} V^\pi_{\mathcal{M}_\tau(x)}(s_0') &\le \mathbb{E}_{a_0\sim\pi(s_0')}\Big[r(s_0',a_0) - \tau h_{s_0'}(\pi(s_0')) + \gamma \mathbb{E}_{s_1\sim\mathcal{P}_x(s_0',a_0)}\big[V^*_{\mathcal{M}_\tau(x)}(s_1)\big]\Big] \\ &< \mathbb{E}_{a_0\sim\pi_y^*(s_0')}\Big[r(s_0',a_0) - \tau h_{s_0'}(\pi_y^*(s_0')) + \gamma \mathbb{E}_{s_1\sim\mathcal{P}_x(s_0',a_0)}\big[V^*_{\mathcal{M}_\tau(x)}(s_1)\big]\Big] \\ &= V^*_{\mathcal{M}_\tau(x)}(s_0') \end{aligned} \tag{B.25}$$

where the last inequality follows from the strong convexity of h and the definition of π_y^*; and the last equality follows from π_y^* is the optimal policy. This proves the lemma.

More generally for $\epsilon \ge 0$, $\mathcal{BM}_\epsilon$ is an ϵ-approximate problem of $\mathcal{BM}$ in a sense that: 1) We have any feasible policy y_ϵ of $\mathcal{BM}_\epsilon$ is ϵ-feasible for $\mathcal{BM}$:

$$\|y_\epsilon - y^*\|^2 \le \tau^{-1}\big(g(x,y_\epsilon) - v(x)\big) \le \tau^{-1}\epsilon \tag{B.26}$$

where $y^* = \pi_y^*$ is the optimal policy, and the first inequality follows from τ-strong-convexity of $g(x,\cdot)$.

2) Moreover, if $f(x,\cdot)$ is L-Lipschitz-continuous with some constant L, the optimal objective value of $\mathcal{BM}$ and $\mathcal{BM}_\epsilon$ are close. Let f^* and $f_\epsilon^* = f(x_\epsilon^*, y_\epsilon^*)$ respectively be the optimal objective value of $\mathcal{BM}$ and $\mathcal{BM}_\epsilon$. Then we have

$$f(x_\epsilon^*, \mathcal{Y}(x_\epsilon^*)) - f(x_\epsilon^*, y_\epsilon^*) \le L\|y_\epsilon^* - \mathcal{Y}(x_\epsilon^*)\| \le L\sqrt{\tau^{-1}\epsilon} \tag{B.27}$$

where the last inequality follows from (B.26) with the fact that under $x_\epsilon^*, y_\epsilon^*$ is a feasible policy of $\mathcal{BM}_\epsilon$ and $\mathcal{Y}(x_\epsilon^*)$ is the optimal policy. The it follows from the fact that $f(x_\epsilon^*, \mathcal{Y}(x_\epsilon^*)) \ge f^*$ and $f(x_\epsilon^*, y_\epsilon^*) \le f^*$, we have

$$|f^* - f(x_\epsilon^*, y_\epsilon^*)| \le f(x_\epsilon^*, \mathcal{Y}(x_\epsilon^*)) - f(x_\epsilon^*, y_\epsilon^*) \le L\sqrt{\tau^{-1}\epsilon}.$$

This concludes the discussion. □

C. Proof in Section 4

C.1. Proof of Lemma 6

We first introduce a generalized Danskin's theorem as follows.

Lemma 13 (Generalized Danskin's Theorem (Clarke, 1975)). *Let $\mathcal{F}$ be a compact set and let a continuous function $\ell : \mathbb{R}^d \times \mathcal{F} \mapsto \mathbb{R}$ satisfy: 1) $\nabla_x \ell(x, y)$ is continuous in (x, y); and 2) given any x, for any $y, y' \in \operatorname{argmax}_{y\in\mathcal{F}} \ell(x, y)$, $\nabla_x \ell(x, y) = \nabla_x \ell(x, y')$. Then let $h(x) \coloneqq \max_{y\in\mathcal{F}} \ell(x, y)$, we have $\nabla h(x) = \nabla_x \ell(x, y^*)$ for any $y^* \in \operatorname{argmax}_{y\in\mathcal{F}} \ell(x, y)$.*

Lemma 13 first follows (Clarke, 1975, Theorem 2.1) where conditions (a)–(d) are guaranteed by Lemma 13's condition 1). Then by (Clarke, 1975, Theorem 2.1 (4)) that we have the Clarke's generalized gradient set of $h(x) = \max_{y\in\mathcal{F}} \ell(x, y)$ is the convex hull of $\{\nabla_x \ell(x, y), y \in \operatorname{argmax}_{y\in\mathcal{F}} \ell(x, y)\}$. It then follows from Lemma 13's condition 2) that this generalized gradient set is a singleton $\{\nabla_x \ell(x, y^*)\}$ with any $y^* \in \operatorname{argmax}_{y\in\mathcal{F}} \ell(x, y)$. Finally it follows from (Clarke, 1975, Proposition 1.13) that $h(x)$ is differentiable with gradient $\nabla_x \ell(x, y^*)$.

Now to prove Lemma 6, it suffices to prove $\nabla \max_{y\in\mathcal{Y}} V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho) = \nabla_x V^{\pi_{y^*}}_{\mathcal{M}_\tau(x)}(\rho)|_{y^*\in\mathcal{Y}^*(x)}$. This arguments is true following from Assumption 1 and the generalized Danskin's theorem above, with $\ell(x, y) = V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho)$.

C.2. Proof of Lemma 7

Proof. **(a).** Under the assumptions in (a), Lemma 6 holds. It then follows from

$$\nabla_x V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho) = \mathbb{E}\big[\sum_{t=0}^{\infty} \gamma^t \nabla r_x(s_t, a_t) | s_0 \sim \rho, \pi_y\big] \tag{C.1}$$

that the result holds.

(b). Given the follower's policy π_y, define the Stackelberg MDP from the leader's view as

$$\mathcal{M}(\pi_y) = \{\mathcal{S}, \mathcal{A}_l, r_{\pi_y}(s, a_l) = \mathbb{E}_{a_f\sim\pi_y(s)}[r_f(s, a_f, a_l)] - \tau h_{f,s}(\pi_y(s)), \mathcal{P}_{\pi_y}(\cdot|s, a_l) = \mathbb{E}_{a_f\sim\pi_y(s)}[\mathcal{P}(\cdot|s, a_l, a_f)]\}$$

Note $\mathcal{M}(\pi_y)$ does not include a regularization for its policy π_x. By Lemma 12, we have the follower's value function $V_f^{\pi_x,\pi_y}(s)$ can be rewritten from the viewpoint that π_y is the main policy and π_x is part of the follower's MDP, that is, $V_f^{\pi_x,\pi_y}(s) = V^{\pi_y}_{\mathcal{M}_\tau(x)}(s)$. It can be proven similarly that $V_f^{\pi_x,\pi_y}(s) = V^{\pi_x}_{\mathcal{M}(\pi_y)}(s)$. Therefore, we have $V^{\pi_y}_{\mathcal{M}_\tau(x)}(s) = V^{\pi_x}_{\mathcal{M}(\pi_y)}(s)$ and

$$\begin{aligned}\nabla_x V^{\pi_y}_{\mathcal{M}_\tau(x)}(s) &= \nabla_x V^{\pi_x}_{\mathcal{M}(\pi_y)}(s)\\ &= \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t Q^{\pi_x}_{\mathcal{M}(\pi_y)}(s_t, a_{l,t}) \nabla \log \pi_x(a_{l,t}|s_t) \Big| s_0 = s, \pi_x\Big]\end{aligned} \tag{C.2}$$

where the last equality follows from the policy gradient theorem (Sutton et al., 2000). We have

$$\begin{aligned}Q^{\pi_x}_{\mathcal{M}(\pi_y)}(s, a_l) &= r_{\pi_y}(s, a_l) + \gamma \mathbb{E}_{s'\sim\mathcal{P}_{\pi_y}(s,a_l)}[V^{\pi_x}_{\mathcal{M}(\pi_y)}(s')]\\ &= \mathbb{E}_{a_f\sim\pi_y(s)}[r_f(s, a_f, a_l)] - \tau h_{f,s}(\pi_y(s)) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(s,a_l,a_f),a_f\sim\pi_y(s)}[V_f^{\pi_x,\pi_y}(s')]\\ &= \mathbb{E}_{a_f\sim\pi_y(s)}[Q_f^{\pi_x,\pi_y}(s, a_l, a_f)] - \tau h_{f,s}(\pi_y(s))\end{aligned} \tag{C.3}$$

where the last equality follows from the definition of $Q_f^{\pi_x,\pi_y}(s, a_l, a_f)$ in Section 2.2. Substituting the above equality into (C.2) yields

$$\begin{aligned}\nabla_x V^{\pi_y}_{\mathcal{M}_\tau(x)}(s) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t Q_f^{\pi_x,\pi_y}(s_t, a_{l,t}, a_{f,t}) \nabla \log \pi_x(a_{l,t}|s_t) \Big| s_0 = s, \pi_x, \pi_y\Big]\\ - \tau \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t h_{f,s_t}(\pi_y(s_t)) \nabla \log \pi_x(a_{l,t}|s_t) \Big| s_0 = s, \pi_x, \pi_y\Big]\end{aligned} \tag{C.4}$$

It then follows from Lemma 6 that

$$\begin{aligned}\nabla_x p(x,y) &= -\nabla_x V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho) + \nabla_x V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho)|_{\pi_y=\pi_y^*(x)} \\ &= -\mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t\big(Q_f^{\pi_x,\pi_y}(s_t,a_{l,t},a_{f,t}) - \tau h_{f,s_t}(\pi_y(s_t))\big)\nabla\log\pi_x(a_{l,t}|s_t)\big|s_0=s,\pi_x,\pi_y\Big] \\ &\quad + \mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t\big(Q_f^{\pi_x,\pi_y^*(x)}(s_t,a_{l,t},a_{f,t}) - \tau h_{f,s_t}(\pi_y^*(x)(s_t))\big)\nabla\log\pi_x(a_{l,t}|s_t)\big|s_0=s,\pi_x,\pi_y^*(x)\Big] \qquad \text{(C.5)}\end{aligned}$$

where $\pi_y^*(x)$ is the follower's optimal policy given leader's policy π_x. □

C.3. Proof of Lemma 8

Proof. We first consider $\nabla_x g(x,y)$. To prove $\nabla_x g(x,y)$ exist, it suffices to show $\nabla q_{s,a}(x)$ exist for any (s,a). By Lemma 13, to show $q_{s,a}(x) = -\max_{\pi\in\Pi} Q^\pi_{\mathcal{M}_\tau(x)}(s,a)$ is differentiable, it remains to show that $\operatorname{argmax}_{\pi\in\Pi} Q^\pi_{\mathcal{M}_\tau(x)}(s,a)$ is a singleton. By Lemma 4, the optimal policy of $\mathcal{M}_\tau(x)$ is unique. Since the unique optimal policy $\pi_y^*(x) \in \operatorname{argmax}_{\pi\in\Pi} Q^\pi_{\mathcal{M}_\tau(x)}(s,a)$, it suffices to show any policy π different from $\pi_y^*(x)$ leads to $Q^\pi_{\mathcal{M}_\tau(x)}(s,a) < Q^{\pi_y^*(x)}_{\mathcal{M}_\tau(x)}(s,a)$. Next we prove this result.

By the uniqueness of the optimal policy, the policies different from $\pi_y^*(x)$ are non-optimal, that is, for any non-optimal π, there exists state $\bar{s}$ such that $V^\pi_{\mathcal{M}_\tau(x)}(\bar{s}) < V^{\pi_y^*(x)}_{\mathcal{M}_\tau(x)}(\bar{s})$. By the Bellman equation, we have for any T,

$$Q^\pi_{\mathcal{M}_\tau(x)}(s,a) = \mathbb{E}\Big[\sum_{t=0}^{T-1}\gamma^t r_x(s_t,a_t)|\pi, s_0=s, a_0=a\Big] + \gamma^T\mathbb{E}_{s_T\sim P_x^\pi(\cdot|s_0=s,a_0=a)}[V^\pi_{\mathcal{M}_\tau(x)}(s_T)] \qquad \text{(C.6)}$$

By the irreducible Markov chain assumption, there exists i such that $P_x^\pi(s_i=\bar{s}|s_0=s,a_0=a)>0$. Choosing $T=i$ in the above equality yields

$$\begin{aligned}Q^\pi_{\mathcal{M}_\tau(x)}(s,a) &= \mathbb{E}\Big[\sum_{t=0}^{i-1}\gamma^t r_x(s_t,a_t)|\pi, s_0=s, a_0=a\Big] + \gamma^i\mathbb{E}_{s_i\sim P_x^\pi(\cdot|s_0=s,a_0=a)}[V^\pi_{\mathcal{M}_\tau(x)}(s_i)] \\ &< \mathbb{E}\Big[\sum_{t=0}^{i-1}\gamma^t r_x(s_t,a_t)|\pi, s_0=s, a_0=a\Big] + \gamma^i\mathbb{E}_{s_i\sim P_x^\pi(\cdot|s_0=s,a_0=a)}[V^{\pi_y^*(x)}_{\mathcal{M}_\tau(x)}(s_i)] \\ &\le Q^{\pi_y^*(x)}_{\mathcal{M}_\tau(x)}(s,a) \qquad \text{(C.7)}\end{aligned}$$

where the first inequality follows from $V^\pi_{\mathcal{M}_\tau(x)}(\bar{s}) < V^{\pi_y^*(x)}_{\mathcal{M}_\tau(x)}(\bar{s})$ and $P_x^\pi(s_i=\bar{s}|s_0=s,a_0=a)>0$; and the last inequality follows from the optimality of $\pi_y^*(x)$.

Given (C.7), we can conclude that $q_{s,a}(x)$ is differentiable with the gradient

$$\nabla q_{s,a}(x) = -\nabla_x Q^\pi_{\mathcal{M}_\tau(x)}(s,a)|_{\pi=\pi_y^*(x)}. \qquad \text{(C.8)}$$

Then $\nabla_x g(x,y)$ can be computed as

$$\nabla_x g(x,y) = -\mathbb{E}_{s\sim\rho,a\sim\pi_y(s)}\big[\nabla_x Q^\pi_{\mathcal{M}_\tau(x)}(s,a)\big]\big|_{\pi=\pi_y^*(x)}. \qquad \text{(C.9)}$$

Since $g(x,\cdot)$ is smooth and strongly-convex, we can use the Danskins' theorem to obtain

$$\nabla v(x) = \nabla_x g(x,y)|_{y=\operatorname{argmin}_{y\in\mathcal{Y}} g(x,y)} = \nabla_x g(x,y)|_{y=\pi_y^*(x)} \qquad \text{(C.10)}$$

where the last equality follows from Lemma 4. This completes the proof. □

C.4. Proof of Lemma 9

Proof. We first prove the first bullet. We have

$$\nabla_x Q^{\pi}_{\mathcal{M}_\tau(x)}(s,a) = \mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t \nabla_x r_x(s_t,a_t)|\pi, s_0=s, a_0=a\Big]. \tag{C.11}$$

It can be checked that Assumption 2 holds and then $\nabla_x g(x,y)$, $\nabla v(x)$ follow from Lemma 8 with (C.11).

We next prove the second bullet. By (C.4), we have

$$\begin{aligned}\nabla_x V^{\pi_y}_{\mathcal{M}_\tau(x)}(s) &= \mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t Q_f^{\pi_x,\pi_y}(s_t,a_{l,t},a_{f,t})\nabla\log\pi_x(a_{l,t}|s_t)\Big|s_0=s,\pi_x,\pi_y\Big]\\ &\quad-\tau\mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t h_{f,s_t}(\pi_y(s_t))\nabla\log\pi_x(a_{l,t}|s_t)\Big|s_0=s,\pi_x,\pi_y\Big]\end{aligned} \tag{C.12}$$

Then

$$\begin{aligned}\nabla_x Q^{\pi_y}_{\mathcal{M}_\tau(x)}(s,a_f) &= \nabla_x\big(r_x(s,a_f)+\gamma\mathbb{E}_{s'\sim\mathcal{P}_x(s,a_f)}[V^{\pi_y}_{\mathcal{M}_\tau(x)}(s')\big)\\ &= \nabla_x\big(\mathbb{E}_{a_l\sim\pi_x(s)}[r_l(s,a_l,a_f)]+\gamma\mathbb{E}_{a_l\sim\pi_x(s),s'\sim\mathcal{P}(s,a_l,a_f)}[V^{\pi_y}_{\mathcal{M}_\tau(x)}(s')]\big)\end{aligned} \tag{C.13}$$

where the last eqaulity follows from the definition of $\mathcal{M}_\tau(x)$ in Lemma 12. Using the log-trick, we can write

$$\begin{aligned}\nabla_x Q^{\pi_y}_{\mathcal{M}_\tau(x)}(s,a_f) &= \mathbb{E}_{a_l\sim\pi_x(s)}\Big[\big(r(s,a_l,a_f)+\gamma\mathbb{E}_{s'\sim\mathcal{P}(s,a_l,a_f)}[V_f^{\pi_x,\pi_y}(s')]\big)\nabla\log\pi_x(a_l|s)\Big]\\ &\quad+\gamma\mathbb{E}_{a_l\sim\pi_x(s),s'\sim\mathcal{P}(s,a_l,a_f)}[\nabla_x V^{\pi_y}_{\mathcal{M}_\tau(x)}(s')]\end{aligned} \tag{C.14}$$

Substituting (C.4) into the above equality yields

$$\begin{aligned}\nabla_x Q^{\pi_y}_{\mathcal{M}_\tau(x)}(s,a_f) &= \mathbb{E}_{a_l\sim\pi_x(s)}\Big[\big(r(s,a_l,a_f)+\gamma\mathbb{E}_{s'\sim\mathcal{P}(s,a_l,a_f)}[V_f^{\pi_x,\pi_y}(s')]\big)\nabla\log\pi_x(a_l|s)\Big]\\ &\quad+\gamma\mathbb{E}_{a_l\sim\pi_x(s),s'\sim\mathcal{P}(s,a_l,a_f)}\mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t Q_f^{\pi_x,\pi_y}(s_t,a_{l,t},a_{f,t})\nabla\log\pi_x(a_{l,t}|s_t)\Big|s_0=s',\pi_x,\pi_y\Big]\\ &\quad-\tau\gamma\mathbb{E}_{a_l\sim\pi_x(s),s'\sim\mathcal{P}(s,a_l,a_f)}\mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t h_{f,s_t}(\pi_y(s_t))\nabla\log\pi_x(a_{l,t}|s_t)\Big|s_0=s',\pi_x,\pi_y\Big]\end{aligned}$$

Using the definition of $Q_f^{\pi_x,\pi_y}$ in the first term, and taking γ of the second and third term inside the expectation gives

$$\begin{aligned}\nabla_x Q^{\pi_y}_{\mathcal{M}_\tau(x)}(s,a_f) &= \mathbb{E}_{a_l\sim\pi_x(s)}\big[Q_f^{\pi_x,\pi_y}(s,a_l,a_f)\nabla\log\pi_x(a_l|s)\big]\\ &\quad+\mathbb{E}_{a_l\sim\pi_x(s),s'\sim\mathcal{P}(s,a_l,a_f)}\mathbb{E}\Big[\sum_{t=1}^{\infty}\gamma^t Q_f^{\pi_x,\pi_y}(s_t,a_{l,t},a_{f,t})\nabla\log\pi_x(a_{l,t}|s_t)\Big|s_1=s',\pi_x,\pi_y\Big]\\ &\quad-\tau\mathbb{E}_{a_l\sim\pi_x(s),s'\sim\mathcal{P}(s,a_l,a_f)}\mathbb{E}\Big[\sum_{t=1}^{\infty}\gamma^t h_{f,s_t}(\pi_y(s_t))\nabla\log\pi_x(a_{l,t}|s_t)\Big|s_1=s',\pi_x,\pi_y\Big]\end{aligned}$$

Continuing from above, combining the first and second term yields

$$\begin{aligned}\nabla_x Q^{\pi_y}_{\mathcal{M}_\tau(x)}(s,a_f) &= \mathbb{E}_{a_l\sim\pi_x(s)}\mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t Q_f^{\pi_x,\pi_y}(s_t,a_{l,t},a_{f,t})\nabla\log\pi_x(a_{l,t}|s_t)\Big|s_0=s,a_{l,0}=a_l,a_{f,0}=a_f,\pi_x,\pi_y\Big]\\ &\quad-\tau\mathbb{E}_{a_l\sim\pi_x(s),s'\sim\mathcal{P}(s,a_l,a_f)}\mathbb{E}\Big[\sum_{t=1}^{\infty}\gamma^t h_{f,s_t}(\pi_y(s_t))\nabla\log\pi_x(a_{l,t}|s_t)\Big|s_1=s',\pi_x,\pi_y\Big]\\ &= \mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t Q_f^{\pi_x,\pi_y}(s_t,a_{l,t},a_{f,t})\nabla\log\pi_x(a_{l,t}|s_t)\Big|s_0=s,a_{f,0}=a_f,\pi_x,\pi_y\Big]\\ &\quad-\tau\mathbb{E}_{a_l\sim\pi_x(s),s'\sim\mathcal{P}(s,a_l,a_f)}\mathbb{E}\Big[\sum_{t=1}^{\infty}\gamma^t h_{f,s_t}(\pi_y(s_t))\nabla\log\pi_x(a_{l,t}|s_t)\Big|s_1=s',\pi_x,\pi_y\Big]\end{aligned} \tag{C.15}$$

It can then be checked that Assumption 2 holds and the result follows from Lemma 8 and (C.15). □

C.5. Sufficient conditions of the smoothness assumption

Lemma 14. *Suppose the following conditions hold.*

(a) For any (s,a), the policy parameterization π_y satisfies 1) $\sum_a \|\nabla \pi_y(a|s)\| \le B_\pi$; and, 2) $\pi_y(a|s)$ is L_y-Lipschitz-smooth.

(b) If $\tau > 0$ then: 1) for any s, assume $|h_s(\pi_y(s))| \le B_h$ and $\|\nabla_y h_s(\pi_y(s))\| \le B_h'$ on $\mathcal{Y}$; and, 2) $h_s(\pi_y(s))$ is L_h-Lipschitz-smooth on $\mathcal{Y}$.

(c) For any (s,a,s'), we have for any $x \in \mathcal{X}$ that 1) $|r_x(s,a)| \le B_r$; and, 2) $V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho)$ is L_{vx}-Lipschitz-smooth on $\mathcal{X}$ uniformly for $y \in \mathcal{Y}$.

Then it holds for any s that $V^{\pi_y}_{\mathcal{M}_\tau(x)}(s)$ is Lipschitz-smooth on $\mathcal{X}\times\mathcal{Y}$:

$$\|\nabla V^{\pi_y}_{\mathcal{M}_\tau(x)}(s) - \nabla V^{\pi_{y'}}_{\mathcal{M}_\tau(x)}(s)\| \le \max\{L_{vx}, L_{vy}\}\|(x,y)-(x',y')\|, \ \ \forall x,x' \in \mathcal{X} \text{ and } y,y' \in \mathcal{Y} \tag{C.16}$$

where $L_{vy} = \mathcal{O}\Big(\frac{B_\pi^2(B_r+\tau B_h)}{(1-\gamma)^3} + \frac{\tau B_h' B_\pi + |\mathcal{A}| L_y (B_r+\tau B_h)}{(1-\gamma)^2} + \frac{\tau(B_h'+L_h)}{1-\gamma}\Big)$.

Condition (a) holds for direct parameterization, where $\sum_a \|\nabla\pi_y(a|s)\| \le |\mathcal{A}|$ and $L_y = 0$; and it also holds for softmax parameterization where $\sum_a \|\nabla\pi_y(a|s)\| = \sum_a \pi_y(a|s)\|\nabla \log \pi_y(a|s)\| \le 1$ and $L_y = 2$. Condition (b) holds for smooth composite of regularization function and policy, e.g., softmax and entropy (Mei et al., 2020, Lemma 14), or direct policy with a smooth regularization. function. Condition (c) 1) is guaranteed since $\mathcal{X}$ is compact and r_x is continuous, and 2) needs to be checked for specific applications. For example, in RLHF/Reward shaping, it can be checked from the formula of $\nabla_x V^{\pi_y}_{\mathcal{M}_\tau(x)}(s)$ in Lemma 7 that there exists $L_{vx} = \frac{L_r}{1-\gamma}$ if r_x is L_r-Lipschitz-smooth.

Proof. We start the proof by showing $V^{\pi_y}_{\mathcal{M}_\tau(x)}(s)$ is Lipschitz-smooth in y on uniformly for any x, that is

$$\|\nabla_y V^{\pi_y}_{\mathcal{M}_\tau(x)}(s) - \nabla_y V^{\pi_{y'}}_{\mathcal{M}_\tau(x)}(s)\| \le L_{vy}\|y-y'\| \tag{C.17}$$

where L_{vy} is a constant independent of x. By the regularized policy gradient derived in (B.10), we have

$$\nabla_y V^{\pi_y}_{\mathcal{M}_\tau(x)}(s) = \frac{1}{1-\gamma}\mathbb{E}_{\bar{s}\sim d^{\pi_y}_{s,x}}\Big[\sum_a Q^{\pi_y}_{\mathcal{M}_\tau(x)}(\bar{s},a)\nabla\pi_y(a|\bar{s})\Big] + \frac{\tau}{1-\gamma}\mathbb{E}_{\bar{s}\sim d^{\pi}_{s,x}}[-\nabla_y h_{\bar{s}}(\pi_y(\bar{s}))] \tag{C.18}$$

where $d^{\pi_y}_{s,x}(\bar{s}) := (1-\gamma)\sum_{t=0}^\infty \gamma^t P_x^{\pi_y}(s_t=\bar{s}|s_0=s)$ is the discounted visitation distribution, and recall $P_x^{\pi_y}(s_t=\bar{s}|s_0=s)$ is the probability of reaching state $\bar{s}$ at time step t under $\mathcal{P}_x$ and π_y. Towards proving (C.17), we prove the following results:

(1) We have $Q^{\pi_y}_{\mathcal{M}_\tau(x)}(s,a)$ is uniformly bounded, and $V^{\pi_y}_{\mathcal{M}_\tau(x)}(s)$ and $Q^{\pi_y}_{\mathcal{M}_\tau(x)}(s,a)$ are Lipschitz continuous in y uniformly for any x.

By the definition of $Q^{\pi_y}_{\mathcal{M}_\tau(x)}(s,a)$, we have

$$|Q^{\pi_y}_{\mathcal{M}_\tau(x)}(s,a)| \le \sum_{t=0}^\infty \gamma^t |r_x(s_t,a_t)| + \tau|h_{s_t}(\pi_y(s_t))| \le \frac{B_r+\tau B_h}{1-\gamma}, \tag{C.19}$$

therefore it follows from (C.18) that

$$\|\nabla_y V^{\pi_y}_{\mathcal{M}_\tau(x)}(s)\| \le B_\pi \frac{B_r+\tau B_h}{(1-\gamma)^2} + \frac{\tau B_h'}{1-\gamma} \tag{C.20}$$

Then by the definition of Q function

$$Q^{\pi_y}_{\mathcal{M}_\tau(x)}(s,a) = r_x(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}_x(s,a)}\big[V^{\pi_y}_{\mathcal{M}_\tau(x)}(s')\big]$$

we have

$$\|\nabla_y Q^{\pi_y}_{\mathcal{M}_\tau(x)}(s,a)\| \le B_r + \gamma\big(B_\pi \frac{B_r+\tau B_h}{(1-\gamma)^2} + \frac{\tau B_h'}{1-\gamma}\big) \tag{C.21}$$

(2) We have $d^{\pi_y}_{s_0,x}(s)$ is Lipschitz-continuous in y uniformly for any x. Define $\mathcal{M}$ as a MDP with $\tau = 0$, $r(s,a) = \mathbf{1}_s$ which is an indicator function of s, and transition $\mathcal{P}_x$. Then we can write $d^{\pi_y}_{s_0,x}(s)$ as

$$\begin{aligned} d^{\pi_y}_{s_0,x}(s) &= \sum_{s'\in\mathcal{S}}\sum_{a'\in\mathcal{A}} d^{\pi_y}_{s_0,x}(s')\pi_y(a'|s')\mathbf{1}_s \\ &= \mathbb{E}_{s\sim d^{\pi_y}_{s_0,x}, a\sim\pi_y(s)}[r(s,a)] \\ &= (1-\gamma)V^{\pi_y}_{\mathcal{M}}(s_0) \end{aligned}$$

where the last equality follows from substituting in $d^{\pi_y}_{s_0,x}(s) = (1-\gamma)\sum_{t=0}^\infty \gamma^t P^{\pi_y}_x(s_t = s|s_0)$. It then follows from (C.20) with $\tau = 0$ (since $V^{\pi_y}_{\mathcal{M}}(s_0)$ has $\tau = 0$) that $d^{\pi_y}_{s_0,x}(s)$ is also uniformly Lipschitz continuous with constant B_π:

$$\sup_{s\in\mathcal{S}} \|d^{\pi_y}_{s_0,x}(s) - d^{\pi_{y'}}_{s_0,x}(s)\| \le B_\pi\|y-y'\|. \tag{C.22}$$

To this end, we can decompose the difference as

$$\begin{aligned} &\nabla_y V^{\pi_y}_{\mathcal{M}_\tau(x)}(s) - \nabla_y V^{\pi_{y'}}_{\mathcal{M}_\tau(x)}(s) \\ &= \frac{1}{1-\gamma}\mathbb{E}_{\bar{s}\sim d^{\pi_y}_{s,x}}\Big[\sum_a Q^{\pi_y}_{\mathcal{M}_\tau(x)}(\bar{s},a)\nabla\pi_y(a|\bar{s})\Big] - \frac{1}{1-\gamma}\mathbb{E}_{\bar{s}\sim d^{\pi_{y'}}_{s,x}}\Big[\sum_a Q^{\pi_y}_{\mathcal{M}_\tau(x)}(\bar{s},a)\nabla\pi_y(a|\bar{s})\Big] \\ &+ \frac{1}{1-\gamma}\mathbb{E}_{\bar{s}\sim d^{\pi_{y'}}_{s,x}}\Big[\sum_a Q^{\pi_y}_{\mathcal{M}_\tau(x)}(\bar{s},a)\nabla\pi_y(a|\bar{s})\Big] - \frac{1}{1-\gamma}\mathbb{E}_{\bar{s}\sim d^{\pi_{y'}}_{s,x}}\Big[\sum_a Q^{\pi_{y'}}_{\mathcal{M}_\tau(x)}(\bar{s},a)\nabla\pi_y(a|\bar{s})\Big] \\ &+ \frac{1}{1-\gamma}\mathbb{E}_{\bar{s}\sim d^{\pi_{y'}}_{s,x}}\Big[\sum_a Q^{\pi_{y'}}_{\mathcal{M}_\tau(x)}(\bar{s},a)\nabla\pi_y(a|\bar{s})\Big] - \frac{1}{1-\gamma}\mathbb{E}_{\bar{s}\sim d^{\pi_{y'}}_{s,x}}\Big[\sum_a Q^{\pi_{y'}}_{\mathcal{M}_\tau(x)}(\bar{s},a)\nabla\pi_{y'}(a|\bar{s})\Big] \\ &+ \frac{\tau}{1-\gamma}\mathbb{E}_{\bar{s}\sim d^{\pi_y}_{s,x}}[-\nabla_y h_{\bar{s}}(\pi_y(\bar{s}))] - \frac{\tau}{1-\gamma}\mathbb{E}_{\bar{s}\sim d^{\pi_{y'}}_{s,x}}[-\nabla_y h_{\bar{s}}(\pi_y(\bar{s}))] \\ &+ \frac{\tau}{1-\gamma}\mathbb{E}_{\bar{s}\sim d^{\pi_{y'}}_{s,x}}[-\nabla_y h_{\bar{s}}(\pi_y(\bar{s}))] - \frac{\tau}{1-\gamma}\mathbb{E}_{\bar{s}\sim d^{\pi_{y'}}_{s,x}}[-\nabla_y h_{\bar{s}}(\pi_{y'}(\bar{s}))] \end{aligned}$$

Continuing from the above inequality, we have

$$\begin{aligned} \|\nabla_y V^{\pi_y}_{\mathcal{M}_\tau(x)}(s) - \nabla_y V^{\pi_{y'}}_{\mathcal{M}_\tau(x)}(s)\| \le{}& \frac{1}{1-\gamma} 2\sup_s \|d^{\pi_y}_{s,x}(s) - d^{\pi_{y'}}_{s,x}(s)\| \sup\Big\|\sum_a Q^{\pi_y}_{\mathcal{M}_\tau(x)}(\bar{s},a)\nabla\pi_y(a|\bar{s})\Big\| \\ &+ \frac{1}{1-\gamma}\sup_a \big|Q^{\pi_y}_{\mathcal{M}_\tau(x)}(\bar{s},a) - Q^{\pi_{y'}}_{\mathcal{M}_\tau(x)}(\bar{s},a)\big|\Big\|\sum_a \nabla\pi_y(a|\bar{s})\Big\| \\ &+ \frac{1}{1-\gamma}\mathbb{E}_{\bar{s}\sim d^{\pi_{y'}}_{s,x}}\Big[\sup_a \big|Q^{\pi_{y'}}_{\mathcal{M}_\tau(x)}(\bar{s},a)\big|\sum_a \big\|\nabla\pi_y(a|\bar{s}) - \nabla\pi_{y'}(a|\bar{s})\big\|\Big] \\ &+ \frac{\tau}{1-\gamma} 2\sup_s \|d^{\pi_y}_{s,x}(s) - d^{\pi_{y'}}_{s,x}(s)\| \sup\|\nabla_y h_{\bar{s}}(\pi_y(\bar{s}))\| \\ &+ \frac{\tau}{1-\gamma}\mathbb{E}_{\bar{s}\sim d^{\pi_{y'}}_{s,x}}\big[\|\nabla_y h_{\bar{s}}(\pi_y(\bar{s})) - \nabla_y h_{\bar{s}}(\pi_{y'}(\bar{s}))\|\big]. \end{aligned} \tag{C.23}$$

Then given the assumptions (a), (b) in this lemma, along with the (C.19)–(C.22), we can get

$$\|\nabla_y V^{\pi_y}_{\mathcal{M}_\tau(x)}(s) - \nabla_y V^{\pi_{y'}}_{\mathcal{M}_\tau(x)}(s)\| \le L_{vy}\|y-y'\| \tag{C.24}$$

where $L_{vy} = \mathcal{O}\Big(\frac{B_\pi^2(B_r+\tau B_h)}{(1-\gamma)^3} + \frac{\tau B_h' B_\pi + |\mathcal{A}|L_y(B_r+\tau B_h)}{(1-\gamma)^2} + \frac{\tau(B_h'+L_h)}{1-\gamma}\Big)$. Thus we conclude

$$\begin{aligned} &\|\nabla V^{\pi_y}_{\mathcal{M}_\tau(x)}(s) - \nabla V^{\pi_{y'}}_{\mathcal{M}_\tau(x)}(s)\|^2 \\ &= \|\nabla_y V^{\pi_y}_{\mathcal{M}_\tau(x)}(s) - \nabla_y V^{\pi_{y'}}_{\mathcal{M}_\tau(x)}(s)\|^2 + \|\nabla_x V^{\pi_{y'}}_{\mathcal{M}_\tau(x)}(s) - \nabla_x V^{\pi_{y'}}_{\mathcal{M}_\tau(x')}(s)\|^2 \\ &\le L_{vy}^2\|y-y'\|^2 + L_{vx}^2\|x-x'\|^2 \le \max\{L_{vy}^2, L_{vx}^2\}\|(x,y)-(x',y')\|^2 \end{aligned} \tag{C.25}$$

which proves the result. □

C.6. Proof of Lemma 10

C.6.1. Smoothness of the value penalty

Proof. Under the two assumptions, Lemma 11 holds and thus $\pi_y^*(x)$ is unique and is $\tau^{-1}C_J$-Lipschitz continuous on $\mathcal{X}$. Thus for any $y, y' \in \mathcal{Y}^*(x)$, we have $\pi_y = \pi_{y'} = \pi_y^*(x)$. With Lemma 6, we have

$$\begin{aligned}\|\nabla \max_{y\in\mathcal{Y}} V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho) - \nabla \max_{y\in\mathcal{Y}} V^{\pi_y}_{\mathcal{M}_\tau(x')}(\rho)\| &= \|\nabla V^{\pi}_{\mathcal{M}_\tau(x)}(\rho)|_{\pi=\pi_y^*(x)} - \nabla V^{\pi}_{\mathcal{M}_\tau(x')}(\rho)|_{\pi=\pi_y^*(x')}\| \\ &\le L_v(\|x-x'\| + \|\pi_y^*(x) - \pi_y^*(x')\|) \\ &\le L_v(1+\tau^{-1}C_J)\|x-x'\|. \end{aligned} \tag{C.26}$$

It then follows from $V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho)$ is L_v-Lipschitz smooth that the value penalty is $L_v(2+\tau^{-1}C_J)$-Lipschitz smooth. □

C.6.2. Smoothness of the Bellman penalty

Proof. First note that Lemma 11 holds and thus $\pi_y^*(x)$ is $\tau^{-1}C_J$-Lipschitz continuous on $\mathcal{X}$. We have $p(x,y) = g(x,y) - v(x)$ where

$$g(x,y) \coloneqq \mathbb{E}_{s\sim\rho}[\langle y_s,\, q_s(x)\rangle + \tau h_s(y_s)]. \tag{C.27}$$

By Lemma 8,

$$\nabla_x g(x,y) = -\mathbb{E}_{s\sim\rho, a\sim y_s}\left[\nabla_x Q^{\pi}_{\mathcal{M}_\tau(x)}(s,a)\right]\big|_{\pi=\pi_y^*(x)}. \tag{C.28}$$

Since $\nabla_x Q^{\pi}_{\mathcal{M}_\tau(x)}(s,a)$ is L_v-Lipschitz continuous by the assumption, and $\pi_y^*(x)$ is $\tau^{-1}C_J$-Lipschitz continuous, we have $\nabla_x g(x,y)$ is $L_v(1+\tau^{-1}C_J)$-Lipschitz continuous at $x\in\mathcal{X}$ uniformly for any y. We also have $\nabla_x g(x,y)$ is C_J-Lipschitz continuous at $y\in\Pi$ uniformly for any $x\in\mathcal{X}$. Therefore, we conclude $\nabla_x g(x,y)$ is $(C_J + L_v(1+\tau^{-1}C_J))$-Lipschitz continuous at (x,y) on $\mathcal{X}\times\Pi$.

Next we have

$$\nabla_y g(x,y) = \Big(\rho(s)q_s(x) + \tau\rho(s)\nabla h_s(y_s)\Big)_{s\in\mathcal{S}}. \tag{C.29}$$

Since q_s is C_J-Lipschitz continuous, and h_s is L_h-Lipschitz smooth, we have $\nabla_y g(x,y)$ is (C_J+L_h)-Lipschitz continuous at (x,y) on $\mathcal{X}\times\Pi$.

Collecting the Lipschitz continuity of $\nabla_x g(x,y)$ and $\nabla_y g(x,y)$ yields $g(x,y)$ is Lipschitz smooth with modulus $L_g = 2C_J + L_v(1+\tau^{-1}C_J) + L_h$. Then we have

$$\|v(x) - v(x')\| = \|g(x,\pi_y^*(x)) - g(x',\pi_y^*(x'))\| \le L_g(\|x-x'\| + \tau^{-1}C_J\|x-x'\|). \tag{C.30}$$

Then we have $p(x,y) = g(x,y) - v(x)$ is Lipschitz smooth with modulus $L_g(2+\tau^{-1}C_J)$. Together with the assumption that f is L_f-Lipschitz smooth gives F_λ is L_v-Lipschitz smooth with $L_v = L_f + \lambda L_g(2+\tau^{-1}C_J)$. □

C.7. Example gradient estimators of the penalty functions

In this section, we give examples of $\hat{\nabla}p(x,y;\hat{\pi})$ that is an estimator of $\nabla p(x,y)$.

Value penalty. Consider choosing the value penalty $p(x,y) = -V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho) + \max_{y\in\mathcal{Y}} V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho)$. Then by Lemma 6, we have

$$\nabla_x p(x,y) = -\nabla_x V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho) + \nabla_x V^{\pi}_{\mathcal{M}_\tau(x)}(\rho)|_{\pi=\pi_y^*(x)}$$

where recall $\pi_y^*(x)$ is the optimal policy of MDP $\mathcal{M}_\tau(x)$ on the policy class $\Pi = \{\pi_y : y\in\mathcal{Y}\}$. A natural choice of $\hat{\nabla}p(x,y;\hat{\pi})$ is then

$$\hat{\nabla}p(x,y;\hat{\pi}) \coloneqq \Big(-\nabla_x V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho) + \nabla_x V^{\pi}_{\mathcal{M}_\tau(x)}(\rho)|_{\pi=\hat{\pi}}, \nabla_y p(x,y)\Big) \tag{C.31}$$

By (Agarwal et al., 2020, Lemma D.3.), there exists constant $L_v = 2\gamma|\mathcal{A}|/(1-\gamma)^3$ that $V^{\pi}_{\mathcal{M}_\tau(x)}(\rho)$ is L_v-Lipschitz-smooth in π for any x. Then the estimation error can be quantified by

$$\|\hat{\nabla} p(x,y;\hat{\pi}) - \nabla p(x,y)\| \le L_v \|\pi_y^*(x) - \hat{\pi}\|. \tag{C.32}$$

Therefore, the estimation error is upper bounded by the policy optimality gap $\|\pi_y^*(x) - \hat{\pi}\|$. One may use efficient algorithms (e.g., policy mirror descent (Zhan et al., 2023)) to solve for $\hat{\pi}$, which has an iteration complexity of $\mathcal{O}(-\log\epsilon)$ to achieve $\|\pi_y^*(x) - \hat{\pi}\| \le \epsilon$. Then Assumption 3 is guaranteed with complexity $\mathcal{O}(-\log(\epsilon_{\text{orac}}/\lambda^2))$.

Bellman penalty. Consider choosing the Bellman penalty $p(x,y) = g(x,y) - v(x)$ where recall $g(x,y) = \mathbb{E}_{s\sim\rho}[\langle y_s,\, q_s(x)\rangle + \tau h_s(y_s)]$ and $v(x) = \min_{y\in\mathcal{Y}} g(x,y)$. Then by Lemma 8, we have

$$\begin{aligned}\nabla_x p(x,y) = &-\mathbb{E}_{s\sim\rho, a\sim\pi_y(s)}\left[\nabla_x Q^{\pi}_{\mathcal{M}_\tau(x)}(s,a)\right]\big|_{\pi=\pi_y^*(x)} \\ &+ \mathbb{E}_{s\sim\rho, a\sim\pi(s)}\left[\nabla_x Q^{\pi}_{\mathcal{M}_\tau(x)}(s,a)\right]\big|_{\pi=\pi_y^*(x)}\end{aligned} \tag{C.33}$$

Therefore, a natural choice of $\hat{\nabla} p(x,y;\hat{\pi})$ is then

$$\hat{\nabla} p(x,y;\hat{\pi}) \coloneqq \Big(-\mathbb{E}_{s\sim\rho, a\sim\pi_y(s)}\left[\nabla_x Q^{\hat{\pi}}_{\mathcal{M}_\tau(x)}(s,a)\right] + \mathbb{E}_{s\sim\rho, a\sim\hat{\pi}(s)}\left[\nabla_x Q^{\hat{\pi}}_{\mathcal{M}_\tau(x)}(s,a)\right], \nabla_y p(x,y)\Big) \tag{C.34}$$

It then follows similarly to (C.32) that Assumption 3 is guaranteed with complexity $\mathcal{O}(-\log(\epsilon_{\text{orac}}/\lambda^2))$.

Example algorithms to get $\hat{\pi}$. Finally, we also explicitly write down the update to obtain $\hat{\pi}$ to be self-contained. If we are using policy mirror descent, then at each outer-iteration k, for $i = 1, \ldots T$ where T is the inner iteration number, we run

$$\pi_k^{i+1}(\cdot|s) = \operatorname*{argmin}_{p\in\Pi}\Big\{ -\langle p,\, Q^{\pi_k^i}_{\mathcal{M}_\tau(x)}(s,\cdot)\rangle + \tau h_s(p) + \frac{1}{\eta} D_h(p, \pi_k^i; \xi_k^i)\Big\},\ \text{for any } s\in\mathcal{S} \tag{C.35}$$

where η is a learning rate, D_h is the Bregman divergence, and ξ_k^i is given by

$$\xi_k^{i+1}(s,a) = \frac{1}{1+\eta\tau}\xi_k^i(s,a) + \frac{\eta}{1+\eta\tau} Q^{\pi_k^i}_{\mathcal{M}_\tau(x)}(s,a). \tag{C.36}$$

Finally, we set the last iterate $\pi_k^{T+1}(\cdot|s)$ as the approximate optimal policy $\hat{\pi}_k$. For theoretical reasons, we use this update in the analysis to gain fast rate. While practically our update scheme is not limited to policy mirror descent. As a simple example, the policy gradient based algorithms can also be used:

$$\hat{y}_k^{i+1} = \text{Proj}_{\mathcal{Y}}\left[\hat{y}_k^i + \eta \nabla_{\hat{y}} V^{\pi_{\hat{y}_k^i}}_{\mathcal{M}_\tau(x)}(\rho)\right],\ \text{for } i = 1, 2, \ldots, T. \tag{C.37}$$

We use the last iterate as the approximate optimal policy parameter: $\hat{\pi}_k = \pi_{\hat{y}_k^{T+1}}$. In the above update, the policy gradient $\nabla_{\hat{y}} V^{\pi_{\hat{y}_k^i}}_{\mathcal{M}_\tau(x)}(\rho)$ can be estimated by a wide range of algorithms including the basic Reinforce (Baxter & Bartlett, 2001), and the advantage actor-critic (Mnih et al., 2016).

C.8. Proof of Theorem 4.1

Proof. In this proof, we write $z = (x,y)$. Consider choosing either the value penalty or the Bellman penalty, then Lemma 10 holds under the assumptions of this theorem. Therefore, F_λ is L_λ-Lipschitz-smooth with $L_\lambda = L_f + \lambda L_p$. Then by Lipschitz-smoothness of F_λ, it holds that

$$\begin{aligned}F_\lambda(z_{k+1}) &\le F_\lambda(z_k) + \langle \nabla F_\lambda(z_k),\, z_{k+1} - z_k\rangle + \frac{L_\lambda}{2}\|z_{k+1} - z_k\|^2 \\ &\overset{\alpha\le\frac{1}{L_\lambda}}{\le} F_\lambda(z_k) + \langle \hat{\nabla} F_\lambda(z_k;\hat{\pi}_k),\, z_{k+1} - z_k\rangle + \frac{1}{2\alpha}\|z_{k+1} - z_k\|^2 + \langle \nabla F_\lambda(z_k) - \hat{\nabla} F_\lambda(z_k;\hat{\pi}_k),\, z_{k+1} - z_k\rangle.\end{aligned} \tag{C.38}$$

Consider the second term in the RHS of (C.38). It is known that z_{k+1} can be written as

$$z_{k+1} = \arg\min_{z\in\mathcal{Z}} \langle \hat{\nabla} F_\lambda(z_k;\hat{\pi}_k),\, z\rangle + \frac{1}{2\alpha}\|z - z_k\|^2.$$

By the first-order optimality condition of the above problem, it holds that

$$\langle \hat{\nabla} F_\lambda(z_k;\hat{\pi}_k) + \frac{1}{\alpha}(z_{k+1}-z_k),\, z_{k+1}-z\rangle \le 0,\ \forall z\in\mathcal{Z}.$$

Since $z_k\in\mathcal{Z}$, we can choose $z=z_k$ in the above inequality and obtain

$$\langle \hat{\nabla} F_\lambda(z_k;\hat{\pi}_k),\, z_{k+1}-z_k\rangle \le -\frac{1}{\alpha}\|z_{k+1}-z_k\|^2. \tag{C.39}$$

Consider the last term in the RHS of (C.38). By Young's inequality, we first have

$$\begin{aligned}\langle \nabla F_\lambda(z_k) - \hat{\nabla} F_\lambda(z_k;\hat{\pi}_k),\, z_{k+1}-z_k\rangle &\le \alpha\|\nabla F_\lambda(z_k) - \hat{\nabla} F_\lambda(z_k;\hat{\pi}_k)\|^2 + \frac{1}{4\alpha}\|z_{k+1}-z_k\|^2\\ &\le \alpha\lambda^2\|\nabla p(z_k) - \hat{\nabla} p(z_k;\hat{\pi}_k)\|^2 + \frac{1}{4\alpha}\|z_{k+1}-z_k\|^2 \end{aligned}\tag{C.40}$$

Substituting (C.40) and (C.39) into (C.38) and rearranging the resulting inequality yield

$$\frac{1}{4\alpha}\|z_{k+1}-z_k\|^2 \le F_\lambda(z_k) - F_\lambda(z_{k+1}) + \alpha\lambda^2\|\nabla p(z_k) - \hat{\nabla} p(z_k;\hat{\pi}_k)\|^2. \tag{C.41}$$

With $\bar{z}_{k+1}$ defined in (4.2), we have

$$\begin{aligned}\|\bar{z}_{k+1}-z_k\|^2 &\le 2\|\bar{z}_{k+1}-z_{k+1}\|^2 + 2\|z_{k+1}-z_k\|^2\\ &\le 2\alpha^2\|\nabla F_\lambda(z_k) - \hat{\nabla} F_\lambda(z_k;\hat{\pi}_k)\|^2 + 2\|z_{k+1}-z_k\|^2\\ &\le 2\alpha^2\lambda^2\|\nabla p(z_k) - \hat{\nabla} p(z_k;\hat{\pi}_k)\|^2 + 2\|z_{k+1}-z_k\|^2 \end{aligned}\tag{C.42}$$

where the second inequality uses non-expansiveness of $\mathrm{Proj}_{\mathcal{Z}}$.

Together (C.41) and (C.42) imply

$$\|\bar{z}_{k+1}-z_k\|^2 \le 10\alpha^2\lambda^2\|\nabla p(z_k) - \hat{\nabla} p(z_k;\hat{\pi}_k)\|^2 + 8\alpha(F_\lambda(z_k) - F_\lambda(z_{k+1})).$$

Since $p(x,y)\ge 0$, $F_\lambda(z)\ge \inf_{z\in\mathcal{Z}} f(z)$ for any $z\in\mathcal{Z}$. Taking a telescope sum of the above inequality and using $G_\lambda(z_k) = \frac{1}{\alpha}(z_k-\bar{z}_{k+1})$ yield

$$\begin{aligned}\sum_{k=1}^K \|G_\lambda(z_k)\|^2 &\le \frac{8\big(F_\lambda(z_1) - \inf_{z\in\mathcal{Z}} f(z)\big)}{\alpha} + \sum_{k=1}^K 10\lambda^2\|\nabla p(z_k) - \hat{\nabla} p(z_k;\hat{\pi}_k)\|^2\\ &\le \frac{8\big(F_\lambda(z_1) - \inf_{z\in\mathcal{Z}} f(z)\big)}{\alpha} + \sum_{k=1}^K \frac{1}{2}\|G_\lambda(z_k)\|^2 + \frac{K}{2}\epsilon_{\mathrm{orac}} \end{aligned}\tag{C.43}$$

where the last inequality follows from Assumption 3. Rearranging gives

$$\sum_{k=1}^K \|G_\lambda(z_k)\|^2 \le \frac{16\big(F_\lambda(z_1) - \inf_{z\in\mathcal{Z}} f(z)\big)}{\alpha} + K\epsilon_{\mathrm{orac}}. \tag{C.44}$$

This proves the first inequality in this theorem. The result for $\mathcal{OS}$ follows similarly with $F_\lambda(y)$ being L_v-Lipschitz-smooth and $\epsilon_{\mathrm{orac}}=0$ since no oracle is needed. □

D. Additional experiment details

D.1. Stackelberg Markov game

For the independent gradient algorithm, we set the learning rate as 0.1, and both the follower and the leader use Monte-Carlo sampling with trajectory length 5 and batch size 16 to estimate the policy gradient. For the PBRL algorithms, to estimate a near-optimal policy $\hat{\pi}$ at each outer iteration, we run the policy gradient algorithm for T steps at every outer iteration. For PBRL with value penalty, we set learning rate 0.1, penalty constant $\lambda=2$, inner iteration number $T=1$, and we use Monte-Carlo sampling with trajectory length 5 and batch size 16 to estimate the policy gradient. For PBRL with Bellman penalty, we use $\lambda=$ and inner iteration number $T=10$ instead.

D.2. Deep reinforcement learning from human feedback

We conduct our experiments in the Arcade Learning Environment (ALE) (Bellemare et al., 2013) wrapped by OpenAI gymnasium which is also used in (Mnih et al., 2016) and (Christiano et al., 2017).

For the Atari games, we use A2C, which is a synchronous version of (Mnih et al., 2016), as the policy gradient estimator in both DRLHF and PBRL. The policy and the critic shares a common base model: The input is fed through 4 convolutional layers of size 8×8, 5×5, 4×4, 4×4, strides $4, 2, 1, 1$ and number of filters $16, 32, 32, 32$, with ReLU activation. This is followed by a fully connected layer of output size 256 and a ReLU non-linearity. The output of the base model is fed to a fully connected layer with scalar output as critic, and another fully connected layer of action space size as policy. The reward predictor has the same input ($84 \times 84 \times 4$ stacked image) as the actor critic. The input is fed through 4 convolutional layers of size 7×7, 5×5, 3×3, 3×3, strides $3, 2, 1, 1$ with 16 filters each and ReLU activation. It is followed by a fully connected layer of size 64, ReLU activation and another fully connected layer of action space size that gives the reward function. We use random dropout (probability 0.5) between fully connected layers to prevent over-fitting (only in reward predictor). The reward predictor and the policy are trained synchronously. Reward predictor is updated for one epoch every 300 A2C update.

We compare trajectories of 25 time steps. At the start of training, we collect 576 pairs of trajectories and warm-up the reward predictor for 500 epochs. After training starts, we collect 16 new pairs per reward learning epoch. We only keep the last collected 3000 pairs in a buffer.

For policy learning, we have actor-critic learning rate 0.0003, entropy coefficient 0.01, actor-critic batch size 16, initial upper-level loss coefficient 0.001 which decays every 3000 actor-critic gradient steps. We find out that the learning procedure is very sensitive to this coefficient, we generally select this coefficient so that the upper-level loss converges stably; for reward learning, we set reward predictor learning rate 0.0003, reward predictor batch size 64, and the reward predictor is trained for one epoch every 500 actor-critic gradient steps. For Beamrider, we change actor-critic learning rate to 7×10^{-5}.