A Constrained Bi-level Optimization Framework for Constrained Preference-Based Reinforcement Learning

Yue Mao[1] **Siyuan Xu**[2] **Shicheng Liu**[2] **Minghui Zhu**[2]
[1]Department of Mechanical Engineering, Pennsylvania State University
[2]School of Electrical Engineering and Computer Science, Pennsylvania State University
{ypm5140, spx5032, sfl5539, muz16}@psu.edu

Abstract

This paper studies the problem of jointly learning a reward function, a cost function, and a policy from an expert's preferences. We formulate the problem as a constrained bi-level optimization problem, where the upper level infers the reward and cost functions from preferences, while the lower level optimizes a policy to best align with those preferences. To solve this problem, we propose a double-loop algorithm, Constrained Bi-level Optimization for Preference-Based Reinforcement Learning (CB-PbRL), which solves the lower-level optimization problem in the inner loop and the upper-level optimization problem in the outer loop. We establish a theoretical guarantee that CB-PbRL converges at a rate of $\mathcal{O}(1/\sqrt{K})$, and we demonstrate its effectiveness across multiple simulation environments.

1 Introduction

In reinforcement learning (RL), an agent aims to learn a policy that maximizes a cumulative reward in a dynamic environment. Designing an appropriate reward function is crucial to ensure that the learned policy aligns with the desired objective. However, reward engineering is challenging due to the need to capture numerous attributes [1] and the risk of reward hacking [2]. Preference-Based Reinforcement Learning (PbRL) has emerged as a promising approach to address these challenges with preference-based feedback from the expert. Existing PbRL-related algorithms [3, 4, 5, 6, 7, 8, 9, 10, 11] typically follow a three-step iterative process: data collection, reward learning, and policy learning. First, the expert provides preference-based feedback on data generated by the current policy. Second, a reward function is inferred from this feedback, often using a likelihood-based method grounded in the Bradley–Terry model [12]. Third, the agent updates its policy to maximize the cumulative return under the learned reward. These steps are repeated as the updated policy generates new data. By learning a reward function from preference-based feedback, PbRL removes the need for manual reward design. For example, in natural language generation, human preferences over candidate responses replace the impractical task of manually defining rewards for all possible outputs. PbRL has recently found success across diverse domains, including robotics [13], natural language processing [4], and games [14].

When the focus is on preference-based feedback, such as binary comparisons between trajectories, the terms reinforcement learning from human feedback (RLHF) and PbRL are often used interchangeably [15]. Despite recent progress, existing PbRL algorithms face two key limitations. The first is constraint inference. In many real-world scenarios, constraints are intrinsic, yet one-dimensional preference-based feedback is insufficient to distinguish between objectives and constraints. Consider the case where a user poses a question that violates ethical guidelines. An ideal large language model should refuse to answer such a question to satisfy safety constraints. However, from the

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

standpoint of helpfulness, the question remains unaddressed, and the response becomes unhelpful. As discussed in [16], the inherently conflicting factors (safety and helpfulness) arise when humans provide preference feedback. In these situations, it would be preferable for humans to indicate preferences across multiple dimensions, such as safety and helpfulness, rather than being restricted to a single axis of comparison. Otherwise, ambiguity arises, since one-dimensional feedback cannot fully express intentions over competing objectives. Meanwhile, the single reward function learned from one-dimensional preference-based feedback may be insufficient to capture the full spectrum of these conflicting factors. As a result, existing PbRL methods can produce policies that violate these constraints, leading to undesired or unsafe behaviors.

The second limitation is misalignment [17], which occurs when the learned reward function fails to fully capture the expert's intended objective, leading to suboptimal policies. In most PbRL methods, feedback collection, reward learning, and policy optimization are performed iteratively. As a result, the dependence of the learned reward function on preference-based feedback generated by the evolving policy is overlooked. Some existing algorithms [18, 19] typically update the reward function using an expanding dataset. In each iteration, new trajectories from the current policy are added to the dataset. However, the reward function is updated based on batches of trajectories collected in previous iterations, without incorporating feedback on trajectories generated by the current policy. This disconnect allows agents to exploit inaccuracies in the reward function and drift away from the intended behavior, thereby exacerbating misalignment.

Contribution. The contributions of this paper are fourfold. First, we study the constrained PbRL problem where preference-based feedback reflects both optimality and constraint satisfaction. We formulate this problem as a constrained bi-level optimization problem. In particular, the upper level aims to learn both a reward function and a cost function from preference-based feedback. The lower level aims to learn a policy that optimizes the learned reward while satisfying the learned cost. The general approach is to treat the lower-level optimization problem as a constrained RL task. However, the constrained RL problem is inherently non-convex, making hypergradient (i.e., the gradient of the upper-level optimization problem) computation possibly intractable since it requires the inverse Hessian of the lower-level objective function. To overcome this challenge, we employ its dual problem to reformulate the lower-level optimization problem. This reformulation guarantees the existence of the required inverse Hessian while preserving the result policy equivalence to the original constrained RL formulation. In this framework, the upper- and lower-level decision variables are coupled: the expert's feedback used to optimize the reward and cost functions is generated from the data induced by the policy according to the current reward and cost functions. This dependency resolves the misalignment limitation. Second, we propose Constrained Bi-level optimization for Preference-Based Reinforcement Learning (CB-PbRL), an algorithm for solving this constrained bi-level optimization problem. A particular challenge is to compute the hypergradient because it includes the derivative of the lower-level optimal solution, which is possibly non-differentiable. We overcome this by applying the Clarke subdifferential approximation with two gradient evaluations. Third, we provide theoretical guarantees showing that the algorithm converges at a rate of $\mathcal{O}(1/\sqrt{K})$, and we validate the performance gap between the expert policy and the learned policy. Fourth, we demonstrate the effectiveness of our approach through four experiments, showing that CB-PbRL simultaneously resolves the misalignment and constraint inference limitations at once.

Related works Due to the space limit, related works are included in Appendix A.

2 Model and Problem statement

In this section, we introduce our formulation of the constrained PbRL problem.

Markov Decision Process (MDP). Consider a MDP without a reward function, denoted as $MDP\backslash R$ $(S, A, \gamma, H, P_0, P)$, which includes the state space S, the action space A, the discount factor $\gamma \in (0, 1]$, the finite horizon $H \in \mathbb{Z}^+$ and the initial state distribution P_0. The transition function $P(s'|s, a)$ denotes the probability density of transitioning from state $s \in S$ to state $s' \in S$ upon taking action $a \in A$. A policy $\pi(a|s)$ specifies the probability density of selecting action a given state s. A trajectory τ is defined as a sequence of state-action pairs over the horizon, i.e., $\tau = s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}$.

Expert's preference-based feedback. Given any two trajectories, the expert could provide three types of feedback [16]: (i) preference label $y_r \in \{0, 1\}$ comparing optimality, (ii) preference label

$y_c \in \{0, 1\}$ comparing constraint satisfaction, and (iii) classification label $z \in \{1, -1\}$ indicating whether a trajectory violates the constraints. For any trajectory pair (τ_0, τ_1), the label $y_r = 0$ if the expert prefers τ_0 over τ_1 in terms of optimality; otherwise, $y_r = 1$. Similarly, the label $y_c = 1$ if the expert prefers τ_0 over τ_1 in terms of constraint satisfaction; otherwise, $y_c = 0$. For any trajectory τ, the label $z = 1$ if τ violates the constraints and $z = -1$ otherwise. We define the unknown expert's feedback distribution as $h_l(y_r, y_c, z_0, z_1|\tau_0, \tau_1)$ where z_0 and z_1 are the classification labels for trajectories τ_0 and τ_1, respectively.

Problem statement. An agent aims to learn a policy that best aligns with the expert's feedback.

3 Problem formulation and Constrained Bi-level Setup

To learn a policy from expert's preference-based feedback, the agent employs two parametric functions: a reward function $r_\phi : S \times A \to \mathbb{R}$ and a cost function $c_\psi : S \times A \to \mathbb{R}$, parameterized by decision variables $\phi \in \mathbb{R}^{d_\phi}$ and $\psi \in \mathbb{R}^{d_\psi}$, respectively. To mitigate the misalignment issue, the agent requires the expert's feedback on trajectories generated by the policy corresponding to the current reward and cost functions. The agent thus simultaneously learns the reward function parameter ϕ, the cost function parameter ψ, and the associated policy. We formulate this problem as a constrained bi-level optimization problem where the lower-level optimization problem learns a policy via constrained RL under the current r_ϕ and c_ψ and the upper-level optimization problem updates r_ϕ and c_ψ by minimizing a cross-entropy loss with the expert's feedback.

The lower-level optimization problem. Define the cumulative reward of a policy π under reward function r_ϕ as $J_{r_\phi}(\pi) \triangleq E^\pi[\sum_{h=0}^{H-1} \gamma^h r_\phi(s_h, a_h)]$, where the initial state is sampled from the distribution P_0. Analogously, define the cumulative cost as $J_{c_\psi}(\pi) \triangleq E^\pi[\sum_{h=0}^{H-1} \gamma^h c_\psi(s_h, a_h)]$. Given r_ϕ and c_ψ, the agent aims to solve the following constrained RL problem:

$$\max_{\pi} \quad J_{r_\phi}(\pi) + \mathcal{H}(\pi), \quad \text{s.t.} \quad J_{c_\psi}(\pi) \leq 0, \tag{1}$$

where $\mathcal{H}(\pi) \triangleq E^\pi[\sum_{h=0}^{H-1} -\gamma^h \ln(\pi(a_h|s_h))]$ is the discounted entropy. The problem (1) is non-convex because both the objective function and the constraint are non-convex. Since the primal problem (1) is difficult to solve due to its non-convexity, we instead derive its dual formulation:

$$\min_{\lambda \geq 0} G(\lambda; \phi, \psi), \tag{2}$$

where the dual function is defined as $G(\lambda; \phi, \psi) \triangleq \max_\pi J_{r_\phi}(\pi) + \mathcal{H}(\pi) - \lambda J_{c_\psi}(\pi)$ and λ is the Lagrange multiplier. Notably, the dual problem (2) is a convex optimization problem. Let $\lambda^*(\phi, \psi)$ denote the optimal solution to the dual problem (2). With $\lambda^*(\phi, \psi)$ we can get the policy $\pi_{\phi,\psi,\lambda^*(\phi,\psi)} = \arg\max_\pi J_{r_\phi}(\pi) + \mathcal{H}(\pi) - \lambda^*(\phi, \psi) J_{c_\psi}(\pi)$. As shown in the following lemma, this policy solves the primal constrained RL problem (1).

Lemma 3.1. *The policy $\pi_{\phi,\psi,\lambda^*(\phi,\psi)}$ is identical to the optimal policy of the primal problem (1).*

Lemma 3.1 is established by proving that the duality gap between the optimal values of the primal and dual problems is zero. The proof is provided in Appendix D. Since the policy $\pi_{\phi,\psi,\lambda^*(\phi,\psi)}$ coincides with the optimal solution to the primal problem (1), we adopt the dual formulation (2) as the lower-level optimization problem. This allows us to compute both the dual optimal solution $\lambda^*(\phi, \psi)$ and the corresponding policy $\pi_{\phi,\psi,\lambda^*(\phi,\psi)}$.

The upper-level optimization problem. With the policy $\pi_{\phi,\psi,\lambda^*(\phi,\psi)}$ obtained from the lower-level optimization problem, the agent samples trajectories to elicit the expert's feedback. To model preferences, we adopt the Bradley–Terry model, which predicts the expert's choices based on the current reward and cost functions. For any trajectory τ, define the cumulative reward of this trajectory as $J_{r_\phi}(\tau) \triangleq \sum_{h=0}^{H-1} \gamma^h r_\phi(s_h, a_h), s_h, a_h \in \tau$ and analogously, the cumulative cost of this trajectory is $J_{c_\psi}(\tau) \triangleq \sum_{h=0}^{H-1} \gamma^h c_\psi(s_h, a_h), s_h, a_h \in \tau$. Given a trajectory pair τ_0, τ_1, the probabilities that an expert prefers τ_0 over τ_1 in terms of optimality and constraint satisfaction are estimated by the Bradley–Terry model as follows:

$$P_{r_\phi}(\tau_0 \succ \tau_1) = \frac{\exp(J_{r_\phi}(\tau_0))}{\exp(J_{r_\phi}(\tau_0)) + \exp(J_{r_\phi}(\tau_1))} = \sigma(J_{r_\phi}(\tau_0) - J_{r_\phi}(\tau_1)) \text{ for optimality,}$$

$P_{c_\psi}(\tau_0 \succ \tau_1) = \dfrac{\exp(J_{c_\psi}(\tau_0))}{\exp(J_{c_\psi}(\tau_0)) + \exp(J_{c_\psi}(\tau_1))} = \sigma(J_{c_\psi}(\tau_0) - J_{c_\psi}(\tau_1))$ for constraint satisfaction, where $\sigma(x) \triangleq \frac{1}{1+\exp(-x)}$.

The agent employs the reward function r_ϕ to model the expert's preferences over optimality, and the cost function c_ψ to model both the constraint satisfaction and constraint violation classification. Using the expert's trajectory preference feedback together with the Bradley–Terry model, the parameters ϕ and ψ are learned by maximizing the log-likelihood of the observed labels.

The pairwise comparison loss for the reward parameter ϕ, based on the optimality preference label y_r, is defined as:

$$\begin{aligned} L_r(\phi, \lambda^*(\phi,\psi)) \triangleq - E_{(\tau_0,\tau_1,y_r)\sim D(y_r,y_c,z_0,z_1,\tau_0,\tau_1;\pi_{\phi,\psi,\lambda^*(\phi,\psi)})}[(1-y_r) \\ \log\sigma(J_{r_\phi}(\tau_0) - J_{r_\phi}(\tau_1)) + y_r \log\sigma(J_{r_\phi}(\tau_1) - J_{r_\phi}(\tau_0))]. \end{aligned} \tag{3}$$

The sample distribution of trajectories and the collected expert's feedback is

$D(y_r, y_c, z_0, z_1, \tau_0, \tau_1; \pi_{\phi,\psi,\lambda^*(\phi,\psi)}) \triangleq h_l(y_r, y_c, z_0, z_1|\tau_0, \tau_1)\rho(\tau_0; \pi_{\phi,\psi,\lambda^*(\phi,\psi)})\rho(\tau_1; \pi_{\phi,\psi,\lambda^*(\phi,\psi)})$, where $\rho(\tau; \pi_{\phi,\psi,\lambda^*(\phi,\psi)}) \triangleq P_0(s_0)\Pi_{h=0}^{H-1} P(s_{h+1}|s_h, a_h)\pi_{\phi,\psi,\lambda^*(\phi,\psi)}(a_h|s_h)$, $s_h, a_h \in \tau$ is the trajectory distribution induced by the policy $\pi_{\phi,\psi,\lambda^*(\phi,\psi)}$. Notice that $h_l(y_r, y_c, z_0, z_1|\tau_0, \tau_1)$ reflects the expert's feedback distribution, which is realized through sampling. The loss function (3) represents the cross-entropy loss between the predictions based on r_ϕ and the actual expert's preference label y_r. Specifically, when $y_r = 0$, the loss encourages the reward function r_ϕ to assign a higher cumulative reward to the trajectory τ_0; conversely, when $y_r = 1$, it encourages a higher reward for τ_1.

The total loss for the cost parameter ψ, incorporating both the constraint satisfaction preference label y_c and the classification label z, is defined as:

$$\begin{aligned} L_c(\psi, \lambda^*(\phi,\psi)) = - E_{(\tau_0,\tau_1,y_c,z_0,z_1)\sim D(y_r,y_c,z_0,z_1,\tau_0,\tau_1;\pi_{\phi,\psi,\lambda^*(\phi,\psi)})} \\ [\underbrace{(1-y_c)\log\sigma(J_{c_\psi}(\tau_0) - J_{c_\psi}(\tau_1)) + y_c \log\sigma(J_{c_\psi}(\tau_1) - J_{c_\psi}(\tau_0))}_{\text{part 1}} \\ + \underbrace{\log\sigma(z_0 J_{c_\psi}(\tau_0)) + \log\sigma(z_1 J_{c_\psi}(\tau_1))}_{\text{part 2}}]. \end{aligned} \tag{4}$$

Part 1 corresponds to the pairwise comparison loss for the cost parameter ψ, which is similar to equation (3). When $y_c = 0$, the comparison loss encourages the cost function c_ψ to assign a higher cumulative cost to the trajectory τ_0, and when $y_c = 1$, it encourages a higher cost for τ_1. In this way, the loss captures the expert's preferences for trajectories with better constraint satisfaction. Part 2 leverages the additional classification labels z_0, and z_1 to further refine the learning of the cost function. Assume the existence of a baseline trajectory τ_b such that $J_{c_\psi}(\tau_b) = 0$. If a trajectory τ violates the constraint (i.e., $z = 1$), the cost function c_ψ should assign a higher cost to τ and $P_{c_\psi}(\tau \succ \tau_b) = \sigma(J_{c_\psi}(\tau) - J_{c_\psi}(\tau_b)) = \sigma(zJ_{c_\psi}(\tau))$. Conversely, if a trajectory τ does not violate the constraint (i.e., $z = -1$), the baseline trajectory τ_b should be assigned a higher cost and $P_{c_\psi}(\tau_b \succ \tau) = \sigma(J_{c_\psi}(\tau_b) - J_{c_\psi}(\tau)) = \sigma(zJ_{c_\psi}(\tau))$ [16]. Together, these two components enable the cost function to align more accurately with the expert's implicit constraint preferences during training.

The agent aims to find the parameters ϕ and ψ that minimize the loss functions defined in Equations (3) and (4). This leads to the following upper-level optimization problem:

$$\arg\min_{\phi,\psi} F(\phi, \psi, \lambda^*(\phi,\psi)) \triangleq L_r(\phi, \lambda^*(\phi,\psi)) + L_c(\psi, \lambda^*(\phi,\psi)). \tag{5}$$

Constrained bi-level optimization formulation. By coupling the upper-level optimization problem (5) with the lower-level optimization problem (2), we obtain the following constrained bi-level optimization problem:

$$\arg\min_{\phi,\psi} F(\phi, \psi, \lambda^*(\phi,\psi)), \text{ s.t. } \quad \lambda^*(\phi,\psi) = \arg\min_{\lambda \geq 0} \quad G(\lambda; \phi, \psi). \tag{6}$$

The lower-level problem computes the optimal Lagrange multiplier $\lambda^*(\phi,\psi)$ and corresponding policy $\pi_{\phi,\psi,\lambda^*(\phi,\psi)}$ based on the given reward function r_ϕ and cost function c_ψ. The upper-level problem then learns the reward function r_{ϕ^*} and the cost function c_{ψ^*} by minimizing the losses derived from the expert's feedback, which incorporates optimality preferences, constraint satisfaction preferences, and classification labels.

4 Algorithm

In this section, we develop the algorithm CB-PbRL to solve the constrained bi-level optimization problem (6). A key challenge lies in computing the hypergradients $H_\phi(\phi, \psi, \lambda^*(\phi, \psi))$ and $H_\psi(\phi, \psi, \lambda^*(\phi, \psi))$, both of which depend on the lower-level optimal solution $\lambda^*(\phi, \psi)$. The expressions for these two hypergradients are

$$H_\phi(\phi, \psi, \lambda^*(\phi, \psi)) = \nabla_\phi F(\phi, \psi, \lambda^*(\phi, \psi)) + \nabla_\phi \lambda^*(\phi, \psi)^T \nabla_\lambda F(\phi, \psi, \lambda^*(\phi, \psi)), \text{ and} \quad (7)$$

$$H_\psi(\phi, \psi, \lambda^*(\phi, \psi)) = \nabla_\psi F(\phi, \psi, \lambda^*(\phi, \psi)) + \nabla_\psi \lambda^*(\phi, \psi)^T \nabla_\lambda F(\phi, \psi, \lambda^*(\phi, \psi)). \quad (8)$$

To compute these hypergradients, the upper-level objective function $F(\phi, \psi, \lambda^*(\phi, \psi))$ and the mapping $\lambda^*(\phi, \psi)$ need to be continuously differentiable with respect to ϕ and ψ. However, this assumption is restrictive as the lower-level optimization problem in (6) is constrained and $\lambda^*(\phi, \psi)$ may fail to be differentiable [20]. In particular, given any ϕ', if there is a non-differentiable point in its neighborhood $\mathcal{B}(\phi', \epsilon)$, a ball centered at ϕ' with radius ϵ, the gradient descent may break down. In such cases, the updates of ϕ' become non-smooth around the non-differentiable point, as the hypergradient is not well-defined there. This non-differentiability presents a fundamental obstacle for hypergradient computation and algorithm design.

We adopt the Clarke subdifferential framework [21] to overcome this challenge.

Definition 4.1. The Clarke subdifferential [21] of a locally Lipschitz function $f(x) : \mathbb{R}^n \to \mathbb{R}$ at a point x is defined as $\bar{\partial} f(x) \triangleq conv\{s \in \mathbb{R}^n : \exists \{x_i\}_{i=1}^\infty \text{s.t.} x_i \to x, \nabla f(x_i) \to s\}$.

Following Definition 4.1, the subdifferential with respect to ϕ can be approximated by extending the Clarke subdifferential as $conv\{H_\phi(\phi', \psi, \lambda^*(\phi', \psi))|\phi' \in \mathcal{B}(\phi, \epsilon)\}$. For any given ϕ, we can sample all feasible points in the neighborhood $\mathcal{B}(\phi, \epsilon)$, compute the corresponding gradients, and take their convex hull to approximate the subdifferential. If all points in $\mathcal{B}(\phi, \epsilon)$ are differentiable, then the convex hull coincides with the exact hypergradient. Otherwise, the approximation accounts for the influence of non-differentiable points. Analogously, the approximated subdifferential with respect to ψ is $conv\{H_\psi(\phi, \psi', \lambda^*(\phi, \psi'))|\psi' \in \mathcal{B}(\psi, \epsilon)\}$. These approximated subdifferentials can then be used to update the decision variables ϕ and ψ, serving as a substitute for gradient descent with exact hypergradients.

However, computing these hypergradients is computationally expensive. This is primarily because it typically involves computing second-order derivatives of the lower-level objective function, such as, $\nabla_\phi \lambda^*(\phi, \psi) = -\nabla_\lambda^2 G(\lambda; \phi, \psi)^{-1} \nabla_{\phi\lambda}^2 G(\lambda; \phi, \psi)$, which requires evaluating and inverting the second-order Hessian, making the hypergradient computation prohibitively expensive in practice. Moreover, a large number of sampled points further amplifies the computational burden, as a hypergradient needs to be computed at each sampled neighbor. As a result, approximating the hypergradient for all sampled neighbors significantly increases the overall computational cost.

We address this computational challenge in two ways. First, for each ϕ or ψ, we approximate the hypergradient when a non-differentiable point is in the neighborhood $\mathcal{B}(\phi, \epsilon)$ or $\mathcal{B}(\psi, \epsilon)$; otherwise, we compute the exact hypergradient. This selective approach reduces computational cost by avoiding unnecessary approximations. Second, inspired by Equation (5) in [20], we further approximate the hypergradient using only two gradient evaluations, thereby achieving a tractable and efficient computation.

4.1 Hypergradient computation

When $\lambda^*(\phi, \psi)$ is continuously differentiable at ϕ or ψ, we compute $H_\phi(\phi, \psi, \lambda^*(\phi, \psi))$ or $H_\psi(\phi, \psi, \lambda^*(\phi, \psi))$. Otherwise, we approximate the subdifferential at ϕ or ψ.

Certifying differentiability. The constrained lower-level solution mapping $\lambda^*(\phi, \psi)$ need not be differentiable. To identify neighborhoods on which it is continuously differentiable, consider the lower-level Lagrangian $L(\phi, \psi, \lambda, \nu) \triangleq G(\lambda; \phi, \psi) - \nu\lambda$, where ν is the multiplier associated with $\lambda \geq 0$. Proposition 4.2 gives a sufficient condition with respect to ϕ; the argument for ψ is analogous.

Proposition 4.2. *Consider $\phi' \in \mathbb{R}^{d_\phi}$ and $\epsilon > 0$. Let $L_{\lambda,\phi}$ and $L_{\nu,\phi}$ denote the Lipschitz constants of $\lambda^*(\phi, \psi)$ and $\nu(\phi, \psi)$, respectively, on $\mathcal{B}(\phi', \epsilon)$. The lower-level solution mapping $\lambda^*(\phi, \psi)$ is continuously differentiable on $\mathcal{B}(\phi', \epsilon)$ if $\lambda^*(\phi', \psi) > L_{\lambda,\phi}\epsilon$ or $\nu(\phi', \psi) > L_{\nu,\phi}\epsilon$.*

The explicit expressions for $\nu(\phi, \psi)$ and $\lambda^*(\phi, \psi)$ are provided in Appendix E and the proof of Proposition 4.2 is provided in Appendix F. When either inequality is satisfied, no point in the neighborhood $\mathcal{B}(\phi', \epsilon)$ results in $\nu(\phi', \psi) = 0$ and $\lambda^*(\phi', \psi) = 0$. When $\nu(\phi', \psi) = 0$ and $\lambda^*(\phi', \psi) = 0$, the lower-level optimal solution $\lambda^*(\phi', \psi)$ lies exactly on the boundary of the constraint, while the constraint itself does not affect the optimality condition. This creates a fragile balance where even small perturbations in ϕ' can abruptly alter whether the constraint is active in shaping $\lambda^*(\phi', \psi)$. Due to the abrupt nature of this transition, the mapping from ϕ' to $\lambda^*(\phi', \psi)$ becomes non-smooth and non-differentiable.

For implementation, we approximate $L_{\lambda,\phi}$ and $L_{\nu,\phi}$ by choosing $\delta > 0$ such that $\|\nabla_\phi \lambda^*(\phi', \psi)\| + \delta \geq \sup_{\bar{\phi} \in \mathcal{B}(\phi', \epsilon)} \|\nabla_\phi \lambda^*(\bar{\phi}, \psi)\| \geq L_{\lambda,\phi}$, and $\|\nabla_\phi \nu(\phi', \psi)\| + \delta \geq \sup_{\bar{\phi} \in \mathcal{B}(\phi', \epsilon)} \|\nabla_\phi \nu(\bar{\phi}, \psi)\| \geq L_{\nu,\phi}$, where the suprema are taken over points at which the corresponding derivatives exist. If either $\lambda^*(\phi, \psi)$ or $\nu(\phi, \psi)$ is not differentiable at ϕ', we sample a nearby point $\phi'' \in \mathcal{B}(\phi', \epsilon)$ at which both mappings are differentiable and use $\|\nabla_\phi \lambda^*(\phi'', \psi)\| + \delta$ and $\|\nabla_\phi \nu(\phi'', \psi)\| + \delta$ for approximations. Proposition 4.2 and the above discussion, including the estimation procedure, apply analogously to ψ with ϕ held fixed.

Computing exact hypergradients. When the conditions in Proposition 4.2 are satisfied, the exact hypergradients can be directly computed. In this case, we obtain the hypergradients $H_\phi(\phi, \psi, \lambda^*(\phi, \psi))$ and $H_\psi(\phi, \psi, \lambda^*(\phi, \psi))$ from Equation (7) and (8), respectively. These hypergradients are computed by applying the chain rule to differentiate the upper-level objective function $F(\phi, \psi, \lambda^*(\phi, \psi))$ with respect to ϕ and ψ.

Approximating subdifferentials. When the conditions in Proposition 4.2 are violated, the subdifferentials are approximated as follows:

$$\mathcal{S}_\phi = \{\kappa H_\phi(\phi, \psi, \lambda^*(\phi, \psi)) + (1 - \kappa)\nabla_\phi F(\phi, \psi, \lambda^*(\phi, \psi)) | \kappa \in [0, 1]\}, \text{ and} \tag{9}$$

$$\mathcal{S}_\psi = \{\kappa H_\psi(\phi, \psi, \lambda^*(\phi, \psi)) + (1 - \kappa)\nabla_\psi F(\phi, \psi, \lambda^*(\phi, \psi)) | \kappa \in [0, 1]\}. \tag{10}$$

One gradient evaluation corresponds to the exact hypergradient $H_\phi(\phi, \psi, \lambda^*(\phi, \psi))$ or $H_\psi(\phi, \psi, \lambda^*(\phi, \psi))$, while the other is obtained by considering the lower-level optimal solution as zero which reduces $H_\phi(\phi, \psi, \lambda^*(\phi, \psi))$ to $\nabla_\phi F(\phi, \psi, \lambda^*(\phi, \psi))$ or $H_\psi(\phi, \psi, \lambda^*(\phi, \psi))$ to $\nabla_\psi F(\phi, \psi, \lambda^*(\phi, \psi))$. Since each subdifferential is approximated using two gradient evaluations, the convex hull simplifies to a linear combination.

Let $d(\mathcal{S}, \mathcal{S}') \triangleq \min\{\|x - y\| | x \in \mathcal{S}, y \in \mathcal{S}'\}$ denote the distance between two sets $\mathcal{S}$ and $\mathcal{S}'$. We now state Lemma 4.3, which establishes the feasibility of approximating the subdifferential using only two hypergradient evaluations.

Lemma 4.3. *Assume there exists a small $\epsilon > 0$ with a $\phi' \in B(\phi, \epsilon)$ leads to $\lambda^*(\phi', \psi)$ non-differentiable at ϕ'. We have $d(S_\phi, conv\{H_\phi(\phi'', \psi, \lambda^*(\phi'', \psi)) | \phi'' \in \mathcal{B}(\phi, \epsilon), \lambda^*(\phi'', \psi)$ differentiable$\}) \leq \mathcal{O}(\epsilon)$.*

Based on Lemma 4.3, we establish that the approximation error between the estimated subdifferential $\mathcal{S}_\phi$ and $conv\{H_\phi(\phi'', \psi, \lambda^*(\phi'', \psi)) | \phi'' \in \mathcal{B}(\phi, \epsilon), \lambda^*(\phi'', \psi)$ differentiable$\}$ is bounded by $\mathcal{O}(\epsilon)$. An analogous result holds for ψ. Therefore, a convex combination of just two gradient evaluations provides a computationally efficient yet theoretically sound approximation of the subdifferential. This construction guarantees stable updates even in the presence of non-differentiable points.

With this approximation strategy, we are now prepared to present the algorithm for solving the constrained bi-level optimization problem.

4.2 Algorithm statement

The CB-PbRL algorithm is summarized in Algorithm 1. At each iteration k, the agent first partially solves the lower-level optimization problem within an inner loop and then uses this intermediate solution to address the upper-level optimization problem in the outer loop. The next step is to determine whether to compute exact hypergradients or approximate subdifferentials by checking the conditions in Proposition 4.2. For each ϕ^k, if $\lambda^k(\phi^k, \psi^k)$ is continuously differentiable over the neighborhood $\mathcal{B}(\phi^k, \epsilon)$, the hypergradient is computed directly by extending Equation (7). Otherwise, it is approximated using Equation (9). An analogous procedure is applied for ψ^k, using either Equation (8) or Equation (10). Finally, the parameters ϕ^k and ψ^k are updated via gradient descent.

Algorithm 1 Constrained Bi-level Optimization for PbRL (CB-PbRL)

Require: Initial reward parameter ϕ^0, cost parameter ψ^0, multiplier $\lambda^0 \geq 0$, outer iterations number K, step sizes $\{\alpha_k\}, \{\beta_k\}, \{\omega_k\}$, and a positive constant δ.

for $k = 1, \ldots, K$ **do**
  $\epsilon_k \leftarrow \frac{1}{k^2}$
  for $t = 0, \ldots, t_k - 1$ **do**
    Calculate the policy with ϕ^k,ψ^k and λ^t through $\pi_{\phi^k,\psi^k,\lambda^t} = \arg\max_\pi \ J_{r_{\phi^k}}(\pi) + \mathcal{H}(\pi) - \lambda^t J_{c_{\psi^k}}(\pi)$.
    Calculate the gradient $\nabla_\lambda G(\lambda^t; \phi^k, \psi^k) = -J_{c_{\psi^k}}(\pi_{\phi^k,\psi^k,\lambda^t})$.
    $\lambda^{t+1} = \max(0, \lambda^t - \omega_k \nabla_\lambda G(\lambda^t; \phi^k, \psi^k))$.
  end for
  $\lambda^k = \lambda^{t_k - 1}$.
  Calculate the policy $\pi_{\phi^k,\psi^k,\lambda^k}$.
  Sample a fixed number of trajectories with $\pi_{\phi^k,\psi^k,\lambda^k}$ and get the feedback labels based on these trajectories.
  if $\nu(\phi^k, \psi^k) > \big(\|\nabla_\phi \nu(\phi^k, \psi^k)\| + \delta\big)\epsilon_k$ or $\lambda^k(\phi^k, \psi^k) > \big(\|\nabla_\phi \lambda^k(\phi^k, \psi^k)\| + \delta\big)\epsilon_k$ **then**
    Compute hypergradient $H_\phi(\phi^k, \psi^k, \lambda^k)$ through Equation (7) by using sampled trajectories and labels.
    Set $g_\phi^k \leftarrow H_\phi(\phi^k, \psi^k, \lambda^k)$.
  else
    Form the set $S_\phi = \big\{\kappa H_\phi(\phi^k, \psi^k, \lambda^k) + (1-\kappa)\nabla_\phi F(\phi^k, \psi^k, \lambda^k) \ : \ \kappa \in [0,1]\big\}$.
    Set $g_\phi^k \leftarrow \arg\min_{g \in S_\phi} \|g\|$.
  end if
  if $\nu(\phi^k, \psi^k) > \big(\|\nabla_\psi \nu(\phi^k, \psi^k)\| + \delta\big)\epsilon_k$ or $\lambda^k(\phi^k, \psi^k) > \big(\|\nabla_\psi \lambda^k(\phi^k, \psi^k)\| + \delta\big)\epsilon_k$ **then**
    Compute hypergradient $H_\psi(\phi^k, \psi^k, \lambda^k)$ through Equation (8) by using sampled trajectories and labels.
    Set $g_\psi^k \leftarrow H_\psi(\phi^k, \psi^k, \lambda^k)$.
  else
    Form the set $S_\phi = \big\{\kappa H_\psi(\phi^k, \psi^k, \lambda^k) + (1-\kappa)\nabla_\psi F(\phi^k, \psi^k, \lambda^k) \ : \ \kappa \in [0,1]\big\}$.
    Set $g_\psi^k \leftarrow \arg\min_{g \in S_\psi} \|g\|$.
  end if
  $\phi^{k+1} \leftarrow \phi^k - \alpha_k g_\phi^k$.
  $\psi^{k+1} \leftarrow \psi^k - \beta_k g_\psi^k$.
end for
return $(\phi^K, \psi^K, \pi_{\phi^K,\psi^K,\lambda^K})$

5 Analytical result

In this section, we present our analytical results on the convergence rate of the hypergradient and the performance gap between the learned policy and the expert's policy. To facilitate the analysis, we impose the following assumptions on the estimated reward function and the cost functions.

Assumption 5.1. For any $s \in S, a \in A$, the reward function $r_\phi(s,a)$ satisfies $|r_\phi(s,a)| \leq R_{b1}$, $\|\nabla_\phi r_\phi(s,a)\| \leq R_{b2}$, and $\|\nabla_\phi^2 r_\phi(s,a)\| \leq R_{b3}$ for any ϕ. Similarly, the cost function $c_\psi(s,a)$ satisfies $|c_\psi(s,a)| \leq C_{b1}$, $\|\nabla_\psi c_\psi(s,a)\| \leq C_{b2}$, and $\|\nabla_\psi^2 c_\psi(s,a)\| \leq C_{b3}$ for any ψ.

Assumption 5.2. We assume there exists a reward (cost) model class $\mathcal{F}$, the estimated reward function r_ϕ, the estimated cost function c_ψ, the true expert reward function r_e, and the true expert cost function c_e belong to $\mathcal{F}$.

Assumption 5.1 implies that the derivatives of the reward function r_ϕ and the cost function c_ψ are bounded. Such boundedness of higher-order derivatives is a standard assumption and has been widely adopted in bi-level optimization [22, 23], RL [24, 25], and RLHF [9]. Assumption 5.2 further states that all reward and cost functions belong to a common function class, such as neural networks. This assumption has also been employed in prior RLHF studies [26, 6].

Accounting the approximation error in Lemma 4.3, we now establish the convergence rate of the proposed algorithm in the following theorem.

Theorem 5.3. *Suppose Assumption 5.1 holds, by choosing* $\alpha_k, \beta_k \propto \frac{1}{\sqrt{K}}$ *and* $\epsilon = \frac{1}{k^2}$*, the following convergence guarantee holds:*

$$\min\{\|g_\phi\|^2 | g_\phi \in S_\phi\} \leq \mathcal{O}(\frac{1}{\sqrt{K}}), \text{ and } \min\{\|g_\psi\|^2 | g_\psi \in S_\psi\} \leq \mathcal{O}(\frac{1}{\sqrt{K}}).$$

Theorem 5.3 shows that Algorithm 1 converges at a rate of $\mathcal{O}(\frac{1}{\sqrt{K}})$. This result implies that the proposed subdifferential approximation method (Equation (9) and (10)) effectively approximates the true subdifferential.

Let $N(\mathcal{F}, \|\cdot\|_\infty, \eta)$ denote the $\eta-$covering number for the function class $\mathcal{F}$ under the infinity norm $\|\cdot\|_\infty$ and n as the number of preference data collected at each iteration. We define the cumulative reward difference between $\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)}$ and the expert's policy π_e according to the true reward function r_e as $SubOptR(\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)}) \triangleq J_{r_e}(\pi_e) - J_{r_e}(\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)})$.

Theorem 5.4. *Suppose Assumptions 5.1 and 5.2 hold, the following holds with probability at least* $1 - \delta$*:*

$$SubOptR(\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)}) \leq \mathcal{O}(\sqrt{\frac{1}{Kn} \ln(\frac{N(\mathcal{F}, \|\cdot\|_\infty, \frac{1}{Kn})}{\delta})}), \text{ and}$$

$$|J_{c_e}(\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)})| \leq \mathcal{O}(\sqrt{\frac{1}{Kn} \ln(\frac{N(\mathcal{F}, \|\cdot\|_\infty, \frac{1}{Kn})}{\delta})}) \text{ with } J_{c_e}(\pi_e) = 0.$$

Theorem 5.4 demonstrates that the policy $\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)}$ is learned efficiently.

6 Experiment

This section presents the experimental results of Algorithm 1 in two benchmark environments: MuJoCo [27], a widely used physics engine for simulating complex robotic systems, and Meta-World [28], a suite of goal-conditioned robotic manipulation tasks featuring a robotic arm. We evaluate CB-PbRL on seven representative tasks: Walker2D, HalfCheetah, Hopper, Swimmer, and Ant from the MuJoCo environment, and Push and Reach from Meta-World. In each environment, the default reward function is adopted as the ground-truth reward, and a corresponding ground-truth cost function is designed. Detailed descriptions of these environments and the construction of the cost functions are provided in the Appendix L.

We compare the proposed CB-PbRL algorithm against three baseline methods: PEBBLE [18], Safe RLHF [16], and PARL [17]. PEBBLE, a standard PbRL algorithm, suffers from both the constraint inference and misalignment issues. Safe RLHF addresses only the constraint inference problem, while PARL focuses solely on mitigating the misalignment issue. In contrast, CB-PbRL is designed to tackle both limitations simultaneously. During training, human feedback is synthetically generated based on ground-truth reward and cost functions, with a possibility of incorrect feedback. For each trajectory pair, the one with a higher cumulative reward is preferred with respect to optimality, while the one with a lower cumulative cost is preferred for constraint satisfaction. A trajectory is considered to violate the constraint if its cumulative cost exceeds zero.

For each cumulative reward and cumulative cost calculation, we evaluate the policy learned at the current environment step by sampling 100 trajectories. We then compute the average cumulative reward under the ground-truth reward function and the average cumulative cost under the ground-truth cost function. Similarly, each learned policy is assessed over 100 independent trials to estimate the constraint violation rate, defined as the fraction of trials in which the agent violates the constraint. Visualizations and figures can be found in Appendix Section L.

We identify two key limitations in existing PbRL approaches: misalignment and constraint inference. Our proposed CB-PbRL algorithm incorporates a bi-level optimization structure to mitigate the misalignment issue and learns a cost function to handle constraint inference. Conceptually, CB-PbRL can be viewed as an enhanced version of naive PbRL that integrates both bi-level optimization and constraint learning. In contrast, PEBBLE represents a naive PbRL approach, PARL adds

Table 1: Cumulative reward (CR), cumulative cost (CC), and constraint violation rate (CVR) at the final environment step.

ENVIRONMENT	ALGORITHM	CR	CC	CVR
WALKER2D	PEBBLE	451.41 ± 9.29	18.05 ± 2.92	0.83 ± 0.01
	SAFE RLHF	517.24 ± 42.71	7.99 ± 2.18	0.23 ± 0.19
	PARL	510.58 ± 12.13	25.87 ± 8.83	0.82 ± 0.14
	CB-PBRL	617.27 ± 24.56	6.50 ± 3.19	0.28 ± 0.03
HALFCHEETAH	PEBBLE	1028.70 ± 36.78	91.82 ± 1.26	0.98 ± 0.01
	SAFE RLHF	1034.50 ± 10.74	53.44 ± 15.14	0.39 ± 0.01
	PARL	1546.75 ± 204.56	153.58 ± 0.72	0.99 ± 0.01
	CB-PBRL	1753.57 ± 25.97	57.77 ± 1.98	0.39 ± 0.01
HOPPER	PEBBLE	243.01 ± 22.10	16.19 ± 0.18	0.44 ± 0.02
	SAFE RLHF	264.73 ± 9.01	4.51 ± 0.50	0.16 ± 0.05
	PARL	331.45 ± 9.06	25.35 ± 5.41	0.62 ± 0.09
	CB-PBRL	379.18 ± 11.85	0.60 ± 0.50	0.04 ± 0.03
SWIMMER	PEBBLE	13.10 ± 16.31	4.58 ± 0.77	0.19 ± 0.05
	SAFE RLHF	32.50 ± 2.33	2.51 ± 0.86	0.08 ± 0.02
	PARL	34.38 ± 5.25	3.48 ± 1.68	0.16 ± 0.07
	CB-PBRL	35.02 ± 2.05	0.12 ± 0.12	0.01 ± 0.01
ANT	PEBBLE	561.82 ± 50.15	107.23 ± 16.61	0.96 ± 0.02
	SAFE RLHF	583.13 ± 69.71	7.31 ± 2.58	0.32 ± 0.02
	PARL	668.44 ± 99.77	86.34 ± 28.78	0.49 ± 0.23
	CB-PBRL	761.23 ± 69.86	9.44 ± 0.70	0.35 ± 0.05
PUSH	PEBBLE	11.79 ± 2.65	27.91 ± 0.07	0.71 ± 0.01
	SAFE RLHF	7.67 ± 2.46	23.94 ± 0.09	0.64 ± 0.01
	PARL	12.22 ± 1.73	27.63 ± 0.05	0.71 ± 0.01
	CB-PBRL	18.67 ± 4.05	22.77 ± 0.35	0.63 ± 0.02
REACH	PEBBLE	428.75 ± 10.45	24.05 ± 1.75	0.71 ± 0.01
	SAFE RLHF	417.65 ± 32.07	19.84 ± 1.33	0.65 ± 0.04
	PARL	807.91 ± 21.41	24.16 ± 0.48	0.73 ± 0.01
	CB-PBRL	922.62 ± 17.76	20.01 ± 0.32	0.66 ± 0.01

only bi-level optimization, and Safe RLHF incorporates only constraint learning. This distinction allows us to isolate and evaluate the individual contributions of each enhancement. To evaluate the effectiveness of the bi-level optimization component, we compare cumulative rewards between PEBBLE and PARL, as well as between Safe RLHF and CB-PbRL. Similarly, to assess the impact of constraint learning, we compare cumulative costs and constraint violation rates between PEBBLE and Safe RLHF, and between PARL and CB-PbRL. As shown in Table 1, CB-PbRL and PARL achieve higher cumulative rewards than Safe RLHF and PEBBLE, respectively, demonstrating that bi-level optimization effectively mitigates misalignment. Meanwhile, Safe RLHF and CB-PbRL yield lower cumulative costs and violation rates than PEBBLE and PARL, respectively, confirming that constraint learning addresses constraint inference. Overall, these results demonstrate that CB-PbRL simultaneously addresses both the misalignment and constraint inference limitations.

7 Conclusion

We develop a constrained PbRL framework that enables an agent to jointly learn a reward function, a cost function, and a policy from the expert's trajectory preference feedback. This framework simultaneously addresses the misalignment and constraint inference limitations. Building on this formulation, we introduced the CB-PbRL algorithm and established theoretical guarantees, including its convergence rate and the performance gap between the optimal and learned policies. Experiments on MuJoCo and Meta-World environments further validate the effectiveness of CB-PbRL, demonstrating its ability to achieve strong performance while ensuring constraint satisfaction.

8 Acknowledgements

This work is partially supported by the National Science Foundation through grants ECCS 2140175 and ECCS 2549536. We would like to thank the reviewers for their constructive and insightful suggestions.

References

[1] W Bradley Knox, Alessandro Allievi, Holger Banzhaf, Felix Schmitt, and Peter Stone. "Reward (mis) design for autonomous driving". In: *Artificial Intelligence* 316 (2023), p. 103829.

[2] Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. "Defining and characterizing reward gaming". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 9460–9471.

[3] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. "Deep reinforcement learning from human preferences". In: *Advances in neural information processing systems* 30 (2017).

[4] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. "Training language models to follow instructions with human feedback". In: *Advances in neural information processing systems* 35 (2022), pp. 27730–27744.

[5] Ellen Novoseller, Yibing Wei, Yanan Sui, Yisong Yue, and Joel Burdick. "Dueling posterior sampling for preference-based reinforcement learning". In: *Conference on Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 1029–1038.

[6] Xiaoyu Chen, Han Zhong, Zhuoran Yang, Zhaoran Wang, and Liwei Wang. "Human-in-the-loop: Provably efficient preference-based reinforcement learning with general function approximation". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 3773–3793.

[7] Yichong Xu, Ruosong Wang, Lin Yang, Aarti Singh, and Artur Dubrawski. "Preference-based reinforcement learning with finite-time guarantees". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 18784–18794.

[8] Runzhe Wu and Wen Sun. "Making rl with preference-based feedback efficient via randomization". In: *arXiv preprint arXiv:2310.14554* (2023).

[9] Banghua Zhu, Michael Jordan, and Jiantao Jiao. "Principled reinforcement learning with human feedback from pairwise or k-wise comparisons". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 43037–43067.

[10] Dingwen Kong and Lin Yang. "Provably feedback-efficient reinforcement learning via active reward learning". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 11063–11078.

[11] Wenhao Zhan, Masatoshi Uehara, Nathan Kallus, Jason D Lee, and Wen Sun. "Provable offline preference-based reinforcement learning". In: *arXiv preprint arXiv:2305.14816* (2023).

[12] Ralph Allan Bradley and Milton E Terry. "Rank analysis of incomplete block designs: I. the method of paired comparisons". In: *Biometrika* 39.3/4 (1952), pp. 324–345.

[13] Donald Joseph Hejna III and Dorsa Sadigh. "Few-shot preference learning for human-in-the-loop rl". In: *Conference on Robot Learning*. PMLR. 2023, pp. 2014–2025.

[14] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. "Reward learning from human preferences and demonstrations in atari". In: *Advances in neural information processing systems* 31 (2018).

[15] Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. "A survey of reinforcement learning from human feedback". In: (2024).

[16] Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. "Safe rlhf: Safe reinforcement learning from human feedback". In: *arXiv preprint arXiv:2310.12773* (2023).

[17] Souradip Chakraborty, Amrit Singh Bedi, Alec Koppel, Dinesh Manocha, Huazheng Wang, Mengdi Wang, and Furong Huang. "PARL: A unified framework for policy alignment in reinforcement learning". In: *arXiv preprint arXiv:2308.02585* (2023), p. 3.

[18] Kimin Lee, Laura Smith, and Pieter Abbeel. "Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training". In: *arXiv preprint arXiv:2106.05091* (2021).

[19] Jongjin Park, Younggyo Seo, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. "Surf: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning". In: *arXiv preprint arXiv:2203.10050* (2022).

[20] Siyuan Xu and Minghui Zhu. "Efficient gradient approximation method for constrained bilevel optimization". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 10. 2023, pp. 12509–12517.

[21] Frank H Clarke. "Generalized gradients and applications". In: *Transactions of the American Mathematical Society* 205 (1975), pp. 247–262.

[22] Shicheng Liu and Minghui Zhu. "Distributed inverse constrained reinforcement learning for multi-agent systems". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 33444–33456.

[23] Siliang Zeng, Chenliang Li, Alfredo Garcia, and Mingyi Hong. "Maximum-likelihood inverse reinforcement learning with finite-time guarantees". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 10122–10135.

[24] Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. "Neural policy gradient methods: Global optimality and rates of convergence". In: *arXiv preprint arXiv:1909.01150* (2019).

[25] Kaiqing Zhang, Alec Koppel, Hao Zhu, and Tamer Basar. "Global convergence of policy gradient methods to (almost) locally optimal policies". In: *SIAM Journal on Control and Optimization* 58.6 (2020), pp. 3586–3612.

[26] Zihao Li, Zhuoran Yang, and Mengdi Wang. "Reinforcement learning with human feedback: Learning dynamic choices via pessimism". In: *arXiv preprint arXiv:2305.18438* (2023).

[27] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulao, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. "Gymnasium: A standard interface for reinforcement learning environments". In: *arXiv preprint arXiv:2407.17032* (2024).

[28] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning". In: *Conference on robot learning*. PMLR. 2020, pp. 1094–1100.

[29] Aadirupa Saha, Aldo Pacchiano, and Jonathan Lee. "Dueling rl: Reinforcement learning with trajectory preferences". In: *International conference on artificial intelligence and statistics*. PMLR. 2023, pp. 6263–6289.

[30] Han Shen, Zhuoran Yang, and Tianyi Chen. "Principled penalty-based methods for bilevel reinforcement learning and rlhf". In: *arXiv preprint arXiv:2402.06886* (2024).

[31] Siyuan Xu and Minghui Zhu. "Online constrained meta-learning: Provable guarantees for generalization". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 15531–15544.

[32] Fangzhu Ming, Feng Gao, Kun Liu, and Xingqi Li. "A constrained DRL-based bi-level coordinated method for large-scale EVs charging". In: *Applied Energy* 331 (2023), p. 120381.

[33] Hossein Mehdipourpicha and Rui Bo. "Risk-constrained bi-level optimization for virtual bidder bidding strategy in day-ahead electricity markets". In: *2020 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE. 2020, pp. 1–5.

[34] Jane J Ye and Daoli Zhu. "New necessary optimality conditions for bilevel programs by combining the MPEC and value function approaches". In: *SIAM Journal on Optimization* 20.4 (2010), pp. 1885–1905.

[35] Risheng Liu, Xuan Liu, Shangzhi Zeng, Jin Zhang, and Yixuan Zhang. "Value-function-based sequential minimization for bi-level optimization". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[36] Wei Yao, Chengming Yu, Shangzhi Zeng, and Jin Zhang. "Constrained bi-level optimization: Proximal lagrangian value function approach and hessian-free algorithm". In: *arXiv preprint arXiv:2401.16164* (2024).

[37] Stephen Gould, Basura Fernando, Anoop Cherian, Peter Anderson, Rodrigo Santa Cruz, and Edison Guo. "On differentiating parameterized argmin and argmax problems with application to bi-level optimization". In: *arXiv preprint arXiv:1607.05447* (2016).

[38] Ioannis Tsaknakis, Prashant Khanduri, and Mingyi Hong. "An implicit gradient-type method for linearly constrained bilevel problems". In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2022, pp. 5438–5442.

[39] Prashant Khanduri, Ioannis Tsaknakis, Yihua Zhang, Jia Liu, Sijia Liu, Jiawei Zhang, and Mingyi Hong. "Linearly constrained bilevel optimization: A smoothed implicit gradient approach". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 16291–16325.

[40] Quan Xiao, Han Shen, Wotao Yin, and Tianyi Chen. "Alternating projected sgd for equality-constrained bilevel optimization". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2023, pp. 987–1023.

[41] Wanli Shi, Yi Chang, and Bin Gu. "Double momentum method for lower-level constrained bilevel optimization". In: *arXiv preprint arXiv:2406.17386* (2024).

[42] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. "Reinforcement learning with deep energy-based policies". In: *International conference on machine learning*. PMLR. 2017, pp. 1352–1361.

[43] Saeed Ghadimi and Mengdi Wang. "Approximation methods for bilevel programming". In: *arXiv preprint arXiv:1802.02246* (2018).

[44] Bodhisattva Sen. "A gentle introduction to empirical process theory and applications". In: *Lecture Notes, Columbia University* 11 (2018), pp. 28–29.

A Related work

PbRL and RLHF. Contemporary RLHF is grounded in the earlier paradigm of PbRL, PbRL primarily relies on preference-based feedback, such as pairwise comparisons and rankings. RLHF encompasses these forms of feedback and extends further to incorporate a wider variety of feedback types, including implicit cues, corrections, and natural language instructions. As a result, there is considerable overlap between the two paradigms in the literature. In particular, when RLHF systems are trained exclusively using preference-based feedback, the terms RLHF and PbRL are often used interchangeably.

The feedback type in PbRL generally falls into two major classes: ranking and pairwise comparison. Pairwise comparisons can be further divided into two types. The first is action-level preference [9, 10, 11, 26], where the expert indicates which action is preferable given a particular state. The second is trajectory-level preference [3, 5, 6, 7, 8], where the expert specifies which of two trajectories is preferred. In this paper, we adopt trajectory-level preference.

Existing PbRL methods can be broadly categorized into online and offline variants. Online approaches [5, 29, 6, 7, 8, 16] continuously update the reward model and policy through interactive agent-environment interactions, where feedback is collected and incorporated during training. This enables iterative refinement of both the reward function and the agent's policy, improving alignment and mitigating overoptimization. Recent works [17, 30] formulate the PbRL problem as a bi-level optimization problem. In [17], the upper-level problem learns a reward function from the expert's feedback, while the lower-level problem computes the policy with respect to the learned reward. Meanwhile, [30] replaces the lower-level optimization problem with a penalty-based formulation, integrating it directly into the upper-level objective function. Safe RLHF [16] decouples the expert's feedback into two components: helpfulness and harmlessness. This formulation yields unbiased preference signals while balancing two potentially conflicting objectives. In particular, the harmlessness objective can be naturally interpreted as a constraint in the optimization problem. Offline approaches [9, 11, 26] train the reward model and policy on a pre-collected dataset and corresponding experts' feedback. This approach is well-suited for domains where online interaction is costly, unsafe, or impractical.

Constrained bi-level optimization. There has been growing interest in constrained bi-level optimization problems, where the lower-level decision variable is subject to constraints. Such problems arise across diverse domains, including machine learning [31], robotics [32], and economics [33]. Two primary approaches have been proposed to address constrained bi-level optimization problems: value function-based methods [34, 35, 36] and implicit gradient-based methods [37, 38, 39, 40, 20, 41]. The value function-based approach reformulates the original bi-level problem into a single-level problem using the value function. The implicit gradient-based approach solves the constrained lower-level problem using the Lagrangian framework and derives gradients via implicit differentiation. However,

a key limitation of both approaches is the assumption that the lower-level constraints are convex. In this paper, we relax this convexity assumption.

B Notion and notations

We define some equations used in the appendix as follows:

- The causal entropy from state s and action a :
  $H(\pi, s, a) = -\ln(\pi(a|s)) + \gamma \int_{s' \in S} P(s'|s, a) H(\pi, s')ds'$.
- the causal entropy from state s:
  $H(\pi, s) = \int_{a \in A} \pi(a|s) H(\pi, s, a)da$.
- The cumulative reward from state s and action a:
  $J_r(\pi, s, a) = r(s, a) + \gamma \int_{s' \in S} P(s'|s, a) J_r(\pi, s')ds'$.
- the cumulative reward from state s:
  $J_r(\pi, s) = \int_{a \in A} \pi(a|s) J_r(\pi, s, a)da$.
- the cumulative cost from state s and action a:
  $J_c(\pi, s, a) = c(s, a) + \gamma \int_{s' \in S} P(s'|s, a) J_c(\pi, s')ds'$
- the cumulative cost from state s:
  $J_c(\pi, s) = \int_{a \in A} \pi(a|s) J_c(\pi, s, a)da$.

During the proofs, we use some symbols to label the methods and theorems used for deriving the current equation. The symbols and what they represent are as follows:

- The symbol (i) represents chain rule.
- The symbol (ii) represents the linearity of expectation.
- The symbol (iii) represents the closed-form of geometric series.
- The symbol (iv) represents the triangle inequality.
- The symbol (v) represents the hölder's inequality.
- The symbol (vi) represents Taylor's theorem.
- The symbol (vii) means the usage of other equations in this paper.
- The symbol $(viii)$ means keeping expansion.

Based on the idea of soft Q learning [42], the formula of the constrained soft Bellman policy for continuous state-action space is as follows:

$$\pi_{\phi,\psi,\lambda}(a|s) = \frac{\exp(Q^{soft}_{\phi,\psi,\lambda}(s, a))}{\exp(V^{soft}_{\phi,\psi,\lambda}(s))}, \tag{11}$$

$$Q^{soft}_{\phi,\psi,\lambda}(s, a) = r_\phi(s, a) - \lambda c_\psi(s, a) + \gamma \int_{s' \in S} P(s'|s, a) V^{soft}_{\phi,\psi,\lambda}(s')ds', \tag{12}$$

$$V^{soft}_{\phi,\psi,\lambda}(s) = \ln \int_{a \in A} \exp(Q^{soft}_{\phi,\psi,\lambda}(s, a))da. \tag{13}$$

From the expression of $\pi_{\phi,\psi,\lambda}$, we can get $\nabla_\lambda \ln(\pi_{\phi,\psi,\lambda}) = \nabla_\lambda Q^{soft}_{\phi,\psi,\lambda}(s, a) - \nabla_\lambda V^{soft}_{\phi,\psi,\lambda}(s)$, therefore, we can calculate $\nabla_\psi Q^{soft}_{\phi,\psi,\lambda}(s, a)$ and $\nabla_\psi V^{soft}_{\phi,\psi,\lambda}(s)$ separately.

Based on the equation of $V^{soft}(s)$, the gradient $\nabla_\psi V^{soft}(s)$ could be calculated as follows:

$$
\begin{aligned}
&\nabla_\lambda V^{soft}_{\phi,\psi,\lambda}(s),\\
&\overset{(i)}{=} \frac{\int_{a\in A}\nabla_\lambda \exp(Q^{soft}_{\phi,\psi,\lambda}(s,a))da}{\int_{a\in A}\exp(Q^{soft}_{\phi,\psi,\lambda}(s,a))da},\\
&\overset{(vii)}{=} \frac{\int_{a\in A}\exp(Q^{soft}_{\phi,\psi,\lambda}(s,a))\nabla_\lambda Q^{soft}_{\phi,\psi,\lambda}(s,a)da}{\exp(V^{soft}_{\phi,\psi,\lambda}(s))},\\
&\overset{(vii)}{=} \int_{a\in A}\pi_{\phi,\psi,\lambda}(a|s)(-c_\psi(s,a)+\gamma\int_{s'\in S}P(s'|s,a)\nabla_\lambda V^{soft}_{\phi,\psi,\lambda}(s')ds')da,\\
&\overset{(viii)}{=} \int_{a\in A}\pi_{\phi,\psi,\lambda}(a|s)(-c_\psi(s,a)+\gamma\int_{s'\in S}P(s'|s,a)\int_{a'\in A}\pi_{\phi,\psi,\lambda}(a'|s')[-c_\psi(s',a')\\
&+\gamma\int_{s''\in S}P(s''|s',a')\nabla_\lambda V^{soft}_{\phi,\psi,\lambda}(s'')ds'']da'ds')da,\\
&= E^{\pi_{\phi,\psi,\lambda}}[\sum_{h=0}^{H-1}-\gamma^h c_\psi(s_h,a_h)|s_0=s]\\
&= -J_{c_\psi}(\pi_{\phi,\psi,\lambda},s),
\end{aligned}
\tag{14}
$$

where the first (vii) uses the expression in equation (13) and (11), and the second (iii) uses the expression in equation (12).
Similarly, we can get $\nabla_\lambda Q^{soft}_{\phi,\psi,\lambda}(s,a)$ from $Q^{soft}_{\phi,\psi,\lambda}(s,a)$.

$$
\begin{aligned}
&\nabla_\lambda Q^{soft}_{\phi,\psi,\lambda}(s,a),\\
&\overset{(vii)}{=} -c_\psi(s,a)+\gamma\int_{s'\in S}P(s'|s,a)\nabla_\lambda V^{soft}_{\phi,\psi,\lambda}(s')ds')da,\\
&\overset{(viii)}{=} E^{\pi_{\phi,\psi,\lambda}}[\sum_{h=0}^{H-1}-\gamma^h c_\psi(s_h,a_h)|s_0=s,a_0=a]\\
&= -J_{c_\psi}(\pi_{\phi,\psi,\lambda},s,a),
\end{aligned}
\tag{15}
$$

where (vii) uses the expression in equation (12).
By summing the results of equation (14) and (15), we can get $\nabla_\psi \ln(\pi_{\phi,\psi,\lambda})$ as follows:

$$
\begin{aligned}
&\nabla_\lambda \ln(\pi_{\phi,\psi,\lambda}(a|s)) = \nabla_\lambda Q^{soft}_{\phi,\psi,\lambda}(s,a) - \nabla_\lambda V^{soft}_{\phi,\psi,\lambda}(s),\\
&= J_{c_\psi}(\pi_{\phi,\psi,\lambda},s) - J_{c_\psi}(\pi_{\phi,\psi,\lambda},s,a).
\end{aligned}
\tag{16}
$$

C Important gradients

In this section, the necessary gradients for future proofs are calculated.

The gradient of entropy corresponding to λ is as follows:

$$
\begin{aligned}
&\nabla_\lambda H(\pi_{\phi,\psi,\lambda}) \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\lambda \pi_{\phi,\psi,\lambda}(a_0|s_0) H(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0)\nabla_\lambda H(\pi_{\phi,\psi,\lambda}, s_0, a_0) da_0 ds_0 \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\lambda \pi_{\phi,\psi,\lambda}(a_0|s_0) H(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0)\nabla_\lambda(-\ln(\pi_{\phi,\psi,\lambda)}(a_0|s_0) \\
&+ \gamma \int_{s_1\in S} P(s_1|s_0,a_0) H(\pi_{\phi,\psi,\lambda}, s_1) ds_1) da_0 ds_0 \\
&\overset{(viii)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \pi_{\phi,\psi,\lambda}(a_0|s_0)\nabla_\lambda \ln(\pi_{\phi,\psi,\lambda}(a_0|s_0)) H(\pi_{\phi,\psi,\lambda}, s_0, a_0) \\
&- \pi_{\phi,\psi,\lambda}(a_0|s_0)\nabla_\lambda \ln(\pi_{\phi,\psi,\lambda})(a_0|s_0) + \gamma \int_{s_1\in S} P(s_1|s_0,a_0) \int_{a_1\in A} \nabla_\lambda \pi_{\phi,\psi,\lambda}(a_1|s_1) H(\pi_{\phi,\psi,\lambda}, s_1, a_1) \\
&+ \pi_{\phi,\psi,\lambda}(a_1|s_1)\nabla_\lambda H(\pi_{\phi,\psi,\lambda}, s_1, a_1) da_1 ds_1 da_0 ds_0 \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \pi_{\phi,\psi,\lambda}(a_0|s_0)\nabla_\lambda \ln(\pi_{\phi,\psi,\lambda}(a_0|s_0)) H(\pi_{\phi,\psi,\lambda}, s_0, a_0) \\
&- \pi_{\phi,\psi,\lambda}(a_0|s_0)\nabla_\lambda \ln(\pi_{\phi,\psi,\lambda})(a_0|s_0) + \gamma \int_{s_1\in S} P(s_1|s_0,a_0) \int_{a_1\in A} \pi_{\phi,\psi,\lambda}(a_1|s_1)\nabla_\lambda \ln(\pi_{\phi,\psi,\lambda}(a_1|s_1)) \\
&H(\pi_{\phi,\psi,\lambda}, s_1, a_1) + \pi_{\phi,\psi,\lambda}(a_1|s_1)\nabla_\lambda(-\ln(\pi_{\phi,\psi,\lambda}(a_1|s_1)) \\
&+ \gamma \int_{s_2\in S} P(s_2|s_1,a_1) H(\pi_{\phi,\psi,\lambda}, s_2) ds_2) da_1 ds_1 da_0 ds_0 \\
&\overset{(viii)}{=} \int_{s\in S} f(\pi_{\phi,\psi,\lambda}, s) \int_{a\in A} \pi_{\phi,\psi,\lambda}(a|s)\nabla_\lambda \ln(\pi_{\phi,\psi,\lambda}(a|s))(H(\pi_{\phi,\psi,\lambda}, s, a) - 1) da ds
\end{aligned}
\tag{17}
$$

where $f(\pi_{\phi,\psi,\lambda}, s)$ is the state visitation frequency. The gradient of cumulative reward $J_{r_\phi}(\pi_{\phi,\psi,\lambda})$ corresponding to λ is as follow:

$$
\begin{aligned}
&\nabla_\lambda J_{r_\phi}(\pi_{\phi,\psi,\lambda}) \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\lambda \pi_{\phi,\psi,\lambda}(a_0|s_0) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0)\nabla_\lambda J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) da_0 ds_0 \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\lambda \pi_{\phi,\psi,\lambda}(a_0|s_0) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0)\nabla_\lambda r_\psi(s_0, a_0) \\
&+ \gamma \int_{s_1\in S} P(s_1|s_0,a_0) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_1) ds_1) da_0 ds_0 \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\lambda \pi_{\phi,\psi,\lambda}(a_0|s_0) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0)\gamma \int_{s_1\in S} P(s_1|s_0,a_0) \\
&\int_{a_1\in A} \nabla_\lambda \pi_{\phi,\psi,\lambda}(a_1|s_1) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_1, a_1) + \pi_{\phi,\psi,\lambda}(a_1|s_1)\nabla_\lambda J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_1, a_1) da_1 ds_1 da_0 ds_0 \\
&\overset{(viii)}{=} \int_{s\in S} f(\pi_{\phi,\psi,\lambda}, s) \int_{a\in A} \pi_{\phi,\psi,\lambda}(a|s)\nabla_\lambda \ln(\pi_{\phi,\psi,\lambda}(a|s)) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s, a) da ds
\end{aligned}
\tag{18}
$$

Through the same process, we can get $\nabla_\lambda J_{c_\psi}(\pi_{\phi,\psi,\lambda})$ as follows:

$$
\nabla_\lambda J_{c_\psi}(\pi_{\phi,\psi,\lambda}) = \int_{s\in S} f(\pi_{\phi,\psi,\lambda}, s) \int_{a\in A} \pi_{\phi,\psi,\lambda}(a|s)\nabla_\lambda \ln(\pi_{\phi,\psi,\lambda}(a|s)) J_{c_\psi}(\pi_{\phi,\psi,\lambda}, s, a) da ds \tag{19}
$$

The gradients of entropy $H(\pi_{\phi,\psi,\lambda})$ with respect to ϕ is shown as follows:

$$
\begin{aligned}
&\nabla_\phi H(\pi_{\phi,\psi,\lambda}) \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\phi \pi_{\phi,\psi,\lambda}(a_0|s_0) H(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\phi H(\pi_{\phi,\psi,\lambda}, s_0, a_0) da_0 ds_0 \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\phi \pi_{\phi,\psi,\lambda}(a_0|s_0) H(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\phi (-\ln(\pi_{\phi,\psi,\lambda)}(a_0|s_0) \\
&+ \gamma \int_{s_1\in S} P(s_1|s_0,a_0) H(\pi_{\phi,\psi,\lambda}, s_1) ds_1) da_0 ds_0 \\
&\overset{(viii)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\phi \ln(\pi_{\phi,\psi,\lambda}(a_0|s_0)) H(\pi_{\phi,\psi,\lambda}, s_0, a_0) \\
&- \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\phi \ln(\pi_{\phi,\psi,\lambda})(a_0|s_0) + \gamma \int_{s_1\in S} P(s_1|s_0,a_0) \int_{a_1\in A} \nabla_\phi \pi_{\phi,\psi,\lambda}(a_1|s_1) H(\pi_{\phi,\psi,\lambda}, s_1, a_1) \\
&+ \pi_{\phi,\psi,\lambda}(a_1|s_1) \nabla_\phi H(\pi_{\phi,\psi,\lambda}, s_1, a_1) da_1 ds_1 da_0 ds_0 \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\phi \ln(\pi_{\phi,\psi,\lambda}(a_0|s_0)) H(\pi_{\phi,\psi,\lambda}, s_0, a_0) \\
&- \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\phi \ln(\pi_{\phi,\psi,\lambda})(a_0|s_0) + \gamma \int_{s_1\in S} P(s_1|s_0,a_0) \int_{a_1\in A} \pi_{\phi,\psi,\lambda}(a_1|s_1) \nabla_\phi \ln(\pi_{\phi,\psi,\lambda}(a_1|s_1)) \\
&H(\pi_{\phi,\psi,\lambda}, s_1, a_1) + \pi_{\phi,\psi,\lambda}(a_1|s_1) \nabla_\phi (-\ln(\pi_{\phi,\psi,\lambda}(a_1|s_1)) \\
&+ \gamma \int_{s_2\in S} P(s_2|s_1,a_1) H(\pi_{\phi,\psi,\lambda}, s_2) ds_2) da_1 ds_1 da_0 ds_0 \\
&\overset{(viii)}{=} \int_{s\in S} f(\pi_{\phi,\psi,\lambda}, s) \int_{a\in A} \pi_{\phi,\psi,\lambda}(a|s) \nabla_\phi \ln(\pi_{\phi,\psi,\lambda}(a|s)) (H(\pi_{\phi,\psi,\lambda}, s, a) - 1) da ds
\end{aligned}
\tag{20}
$$

The gradients of cumulative reward $J_{r_\phi}(\pi_{\phi,\psi,\lambda})$ with respect to ϕ is shown as follows:

$$
\begin{aligned}
&\nabla_\phi J_{r_\phi}(\pi_{\phi,\psi}) \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\phi \pi_{\phi,\psi,\lambda}(a_0|s_0) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\phi J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) da_0 ds_0 \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\phi \pi_{\phi,\psi,\lambda}(a_0|s_0) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\phi (r_\phi(s_0, a_0) \\
&+ \gamma \int_{s_1\in S} P(s_1|s_0,a_0) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_1) ds_1) da_0 ds_0 \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \pi_{\phi,\psi,\lambda}(a_0|s_0) (\nabla_\phi \ln(\pi_{\phi,\psi,\lambda}(a_0|s_0)) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \nabla_\phi r_\phi(s_0, a_0)) \\
&+ \pi_{\phi,\psi,\lambda}(a_0|s_0) \gamma \int_{s_1\in S} P(s_1|s_0,a_0) \int_{a_1\in A} \nabla_\phi \pi_{\phi,\psi,\lambda}(a_1|s_1) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_1, a_1) \\
&+ \pi_{\phi,\psi,\lambda}(a_1|s_1) \nabla_\phi J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_1, a_1) da_1 ds_1 da_0 ds_0 \\
&\overset{(viii)}{=} \int_{s\in S} f(\pi_{\phi,\psi,\lambda}, s) \int_{a\in A} \pi_{\phi,\psi,\lambda}(a|s) (\nabla_\phi \ln(\pi_{\phi,\psi,\lambda}(a|s)) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s, a) + \nabla_\psi r_\phi(s, a)) da ds
\end{aligned}
\tag{21}
$$

The gradients of cumulative cost $J_{c_\psi}(\pi_{\phi,\psi,\lambda})$ with respect to ϕ is shown as follows:

$$
\begin{aligned}
&\nabla_\phi J_{c_\psi}(\pi_{\phi,\psi,\lambda}) \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\phi \pi_{\phi,\psi,\lambda}(a_0|s_0) J_{c_\psi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\phi J_{c_\psi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) da_0 ds_0 \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\phi \pi_{\phi,\psi,\lambda}(a_0|s_0) J_{c_\psi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\phi (c_\psi(s_0, a_0) \\
&+ \gamma \int_{s_1\in S} P(s_1|s_0, a_0) J_{c_\psi}(\pi_{\phi,\psi,\lambda}, s_1) ds_1) da_0 ds_0 \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\phi \pi_{\phi,\psi,\lambda}(a_0|s_0) J_{c_\psi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0) \gamma \int_{s_1\in S} P(s_1|s_0, a_0) \\
&\int_{a_1\in A} \nabla_\phi \pi_{\phi,\psi,\lambda}(a_1|s_1) J_{c_\psi}(\pi_{\phi,\psi,\lambda}, s_1, a_1) + \pi_{\phi,\psi,\lambda}(a_1|s_1) \nabla_\phi J_{c_\psi}(\pi_{\phi,\psi,\lambda}, s_1, a_1) da_1 ds_1 da_0 ds_0 \\
&\overset{(viii)}{=} \int_{s\in S} f(\pi_{\phi,\psi,\lambda}, s) \int_{a\in A} \pi_{\phi,\psi,\lambda}(a|s) \nabla_\psi \ln(\pi_{\phi,\psi,\lambda}(a|s)) J_{c_\psi}(\pi_{\phi,\psi,\lambda}, s, a) da ds
\end{aligned}
\tag{22}
$$

The gradients of entropy $H(\pi_{\phi,\psi,\lambda})$ with respect to ψ is shown as follows:

$$
\begin{aligned}
&\nabla_\psi H(\pi_{\phi,\psi,\lambda}) \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\psi \pi_{\phi,\psi,\lambda}(a_0|s_0) H(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\psi H(\pi_{\phi,\psi,\lambda}, s_0, a_0) da_0 ds_0 \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\psi \pi_{\phi,\psi,\lambda}(a_0|s_0) H(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\psi (-\ln(\pi_{\phi,\psi,\lambda)}(a_0|s_0) \\
&+ \gamma \int_{s_1\in S} P(s_1|s_0, a_0) H(\pi_{\phi,\psi,\lambda}, s_1) ds_1) da_0 ds_0 \\
&\overset{(viii)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\psi \ln(\pi_{\phi,\psi,\lambda}(a_0|s_0)) H(\pi_{\phi,\psi,\lambda}, s_0, a_0) \\
&- \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\psi \ln(\pi_{\phi,\psi,\lambda})(a_0|s_0) + \gamma \int_{s_1\in S} P(s_1|s_0, a_0) \int_{a_1\in A} \nabla_\psi \pi_{\phi,\psi,\lambda}(a_1|s_1) H(\pi_{\phi,\psi,\lambda}, s_1, a_1) \\
&+ \pi_{\phi,\psi,\lambda}(a_1|s_1) \nabla_\psi H(\pi_{\phi,\psi,\lambda}, s_1, a_1) da_1 ds_1 da_0 ds_0 \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\psi \ln(\pi_{\phi,\psi,\lambda}(a_0|s_0)) H(\pi_{\phi,\psi,\lambda}, s_0, a_0) \\
&- \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\psi \ln(\pi_{\phi,\psi,\lambda})(a_0|s_0) + \gamma \int_{s_1\in S} P(s_1|s_0, a_0) \int_{a_1\in A} \pi_{\phi,\psi,\lambda}(a_1|s_1) \nabla_\psi \ln(\pi_{\phi,\psi,\lambda}(a_1|s_1)) \\
&H(\pi_{\phi,\psi,\lambda}, s_1, a_1) + \pi_{\phi,\psi,\lambda}(a_1|s_1) \nabla_\psi (-\ln(\pi_{\phi,\psi,\lambda}(a_1|s_1)) \\
&+ \gamma \int_{s_2\in S} P(s_2|s_1, a_1) H(\pi_{\phi,\psi,\lambda}, s_2) ds_2) da_1 ds_1 da_0 ds_0 \\
&\overset{(viii)}{=} \int_{s\in S} f(\pi_{\phi,\psi,\lambda}, s) \int_{a\in A} \pi_{\phi,\psi,\lambda}(a|s) \nabla_\psi \ln(\pi_{\phi,\psi,\lambda}(a|s)) (H(\pi_{\phi,\psi,\lambda}, s, a) - 1) da ds
\end{aligned}
\tag{23}
$$

The gradients of cumulative reward $J_{r_\phi}(\pi_{\phi,\psi,\lambda})$ with respect to ψ is shown as follows:

$$
\begin{aligned}
&\nabla_\psi J_{r_\phi}(\pi_{\phi,\psi,\lambda}) \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\psi \pi_{\phi,\psi,\lambda}(a_0|s_0) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\psi J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) da_0 ds_0 \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\psi \pi_{\phi,\psi,\lambda}(a_0|s_0) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\psi (r_\phi(s_0, a_0) \\
&+ \gamma \int_{s_1\in S} P(s_1|s_0, a_0) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_1) ds_1) da_0 ds_0 \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\psi \pi_{\phi,\psi,\lambda}(a_0|s_0) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0) \gamma \int_{s_1\in S} P(s_1|s_0, a_0) \\
&\int_{a_1\in A} \nabla_\psi \pi_{\phi,\psi,\lambda}(a_1|s_1) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_1, a_1) + \pi_{\phi,\psi,\lambda}(a_1|s_1) \nabla_\psi J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s_1, a_1) da_1 ds_1 da_0 ds_0 \\
&\overset{(viii)}{=} \int_{s\in S} f(\pi_{\phi,\psi,\lambda}, s) \int_{a\in A} \pi_{\phi,\psi,\lambda}(a|s) \nabla_\psi \ln(\pi_{\phi,\psi,\lambda}(a|s)) J_{r_\phi}(\pi_{\phi,\psi,\lambda}, s, a) da ds
\end{aligned}
\tag{24}
$$

The gradients of cumulative cost $J_{c_\psi}(\pi_{\phi,\psi,\lambda})$ with respect to ψ is shown as follows:

$$
\begin{aligned}
&\nabla_\psi J_{c_\psi}(\pi_{\phi,\psi}) \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\psi \pi_{\phi,\psi,\lambda}(a_0|s_0) J_{c_\psi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\psi J_{c_\psi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) da_0 ds_0 \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \nabla_\psi \pi_{\phi,\psi,\lambda}(a_0|s_0) J_{c_\psi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \pi_{\phi,\psi,\lambda}(a_0|s_0) \nabla_\psi (c_\psi(s_0, a_0) \\
&+ \gamma \int_{s_1\in S} P(s_1|s_0, a_0) J_{c_\psi}(\pi_{\phi,\psi,\lambda}, s_1) ds_1) da_0 ds_0 \\
&\overset{(i)}{=} \int_{s_0\in S} P_0(s_0) \int_{a_0\in A} \pi_{\phi,\psi,\lambda}(a_0|s_0) (\nabla_\psi \ln(\pi_{\phi,\psi,\lambda}(a_0|s_0)) J_{c_\psi}(\pi_{\phi,\psi,\lambda}, s_0, a_0) + \nabla_\psi c_\psi(s_0, a_0)) \\
&+ \pi_{\phi,\psi,\lambda}(a_0|s_0) \gamma \int_{s_1\in S} P(s_1|s_0, a_0) \int_{a_1\in A} \nabla_\psi \pi_{\phi,\psi,\lambda}(a_1|s_1) J_{c_\psi}(\pi_{\phi,\psi,\lambda}, s_1, a_1) \\
&+ \pi_{\phi,\psi,\lambda}(a_1|s_1) \nabla_\psi J_{c_\psi}(\pi_{\phi,\psi,\lambda}, s_1, a_1) da_1 ds_1 da_0 ds_0 \\
&\overset{(viii)}{=} \int_{s\in S} f(\pi_{\phi,\psi,\lambda}, s) \int_{a\in A} \pi_{\phi,\psi,\lambda}(a|s) (\nabla_\psi \ln(\pi_{\phi,\psi,\lambda}(a|s)) J_{c_\psi}(\pi_{\phi,\psi,\lambda}, s, a) + \nabla_\psi c_\psi(s, a)) da ds
\end{aligned}
\tag{25}
$$

D Proof of Lemma 3.1

Proof. Consider the primal constrained entropy-regularized RL problem

$$
p^* = \max_{\pi\in\Pi} J_{r_\phi}(\pi) + \mathcal{H}(\pi) \quad \text{s.t.} \quad J_{c_\psi}(\pi) \le 0,
$$

where

$$
\mathcal{H}(\pi) = \mathbb{E}^\pi \left[\sum_{h=0}^{H-1} \gamma^h \left(-\ln \pi_h(a_h|s_h) \right) \right].
$$

Since the problem is formulated over a finite horizon, the policy may be time-dependent. We write the generic policy as

$$
\pi = \{\pi_h\}_{h=0}^{H-1},
$$

where each $\pi_h(\cdot|s)$ is a probability density over $\mathcal{A}$ at time step h and state s.

For any fixed $\lambda \ge 0$, define the Lagrangian

$$
L(\pi; \phi, \psi, \lambda) = J_{r_\phi}(\pi) + \mathcal{H}(\pi) - \lambda J_{c_\psi}(\pi).
$$

Equivalently,

$$L(\pi;\phi,\psi,\lambda)=\mathbb{E}^{\pi}\left[\sum_{h=0}^{H-1}\gamma^{h}\left(r_{\phi}(s_h,a_h)-\lambda c_{\psi}(s_h,a_h)-\ln\pi_h(a_h|s_h)\right)\right].$$

Then, for fixed λ, maximizing $L(\pi;\phi,\psi,\lambda)$ over π is an entropy-regularized RL problem with $r_{\phi}(s,a)-\lambda c_{\psi}(s,a)$.

For $h=H-1,\ldots,0$, we have

$$Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a)=r_{\phi}(s,a)-\lambda c_{\psi}(s,a)+\gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}\left[V^{\text{soft}}_{h+1,\phi,\psi,\lambda}(s')\right].$$

For each fixed h and s, the entropy-regularized policy improvement step solves

$$V^{\text{soft}}_{h,\phi,\psi,\lambda}(s)=\max_{\pi_h(\cdot|s)}\left\{\int_{a\in\mathcal{A}}\pi_h(a|s)Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a)da-\int_{a\in\mathcal{A}}\pi_h(a|s)\ln\pi_h(a|s)da\right\},$$

subject to

$$\int_{a\in\mathcal{A}}\pi_h(a|s)da=1.$$

Introduce a Lagrange multiplier $\eta_{h,s}\in\mathbb{R}$ for the normalization constraint. The local Lagrangian is

$$\begin{aligned}\mathcal{L}_{h,s}=&\int_{a\in\mathcal{A}}\pi_h(a|s)Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a)da-\int_{a\in\mathcal{A}}\pi_h(a|s)\ln\pi_h(a|s)da\\&+\eta_{h,s}\left(\int_{a\in\mathcal{A}}\pi_h(a|s)da-1\right).\end{aligned}$$

Taking the first-order condition with respect to $\pi_h(a|s)$ gives

$$Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a)-\ln\pi_h(a|s)-1+\eta_{h,s}=0.$$

Therefore,

$$\ln\pi_h(a|s)=Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a)+\eta_{h,s}-1.$$

Thus,

$$\pi_h(a|s)=\exp\left(Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a)+\eta_{h,s}-1\right).$$

Using the normalization constraint,

$$\int_{a\in\mathcal{A}}\pi_h(a|s)da=1,$$

we obtain

$$\int_{a\in\mathcal{A}}\exp\left(Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a)+\eta_{h,s}-1\right)da=1.$$

Since $\eta_{h,s}$ does not depend on a, this implies

$$\exp(\eta_{h,s}-1)=\frac{1}{\int_{a'\in\mathcal{A}}\exp\left(Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a')\right)da'}.$$

Substituting this back gives

$$\pi_{h,\phi,\psi,\lambda}(a|s)=\frac{\exp\left(Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a)\right)}{\int_{a'\in\mathcal{A}}\exp\left(Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a')\right)da'}.$$

Therefore, for the fixed λ, the induced optimal soft Bellman policy is the sequence

$$\pi_{\phi,\psi,\lambda}=\left\{\pi_{h,\phi,\psi,\lambda}\right\}_{h=0}^{H-1}.$$

Equivalently, we have the soft value function as the log-partition function

$$V^{\text{soft}}_{h,\phi,\psi,\lambda}(s)=\ln\int_{a\in\mathcal{A}}\exp\left(Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a)\right)da.$$

Then

$$\pi_{h,\phi,\psi,\lambda}(a|s) = \exp\left(Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a) - V^{\text{soft}}_{h,\phi,\psi,\lambda}(s)\right).$$

We now verify that each $\pi_{h,\phi,\psi,\lambda}(\cdot|s)$ is a valid continuous action probability density. First,

$$\begin{aligned}\int_{a\in\mathcal{A}} \pi_{h,\phi,\psi,\lambda}(a|s)da &= \int_{a\in\mathcal{A}} \frac{\exp\left(Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a)\right)}{\int_{a'\in\mathcal{A}}\exp\left(Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a')\right)da'}da \\ &= \frac{\int_{a\in\mathcal{A}}\exp\left(Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a)\right)da}{\int_{a'\in\mathcal{A}}\exp\left(Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a')\right)da'} \\ &= 1.\end{aligned}$$

Moreover, since

$$\exp\left(Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a)\right) > 0$$

and

$$\int_{a'\in\mathcal{A}} \exp\left(Q^{\text{soft}}_{h,\phi,\psi,\lambda}(s,a')\right)da' > 0,$$

we have

$$\pi_{h,\phi,\psi,\lambda}(a|s) > 0, \qquad \forall s\in\mathcal{S},\ a\in\mathcal{A}.$$

Therefore, for every $h = 0,\dots,H-1$ and $s\in\mathcal{S}$,

$$\int_{a\in\mathcal{A}} \pi_{h,\phi,\psi,\lambda}(a|s)da = 1$$

and

$$\pi_{h,\phi,\psi,\lambda}(a|s) > 0.$$

Thus, for every h and s, $\pi_{h,\phi,\psi,\lambda}(\cdot|s)$ is a valid continuous action probability density. Consequently, the constrained soft Bellman policy

$$\pi_{\phi,\psi,\lambda} = \{\pi_{h,\phi,\psi,\lambda}\}_{h=0}^{H-1}$$

follows the idea in [42] is a valid finite-horizon Markov policy.

Next, define the dual function

$$G(\lambda;\phi,\psi) = \max_{\pi\in\Pi} L(\pi;\phi,\psi,\lambda) = L(\pi_{\phi,\psi,\lambda};\phi,\psi,\lambda).$$

By the envelope theorem,

$$\nabla_\lambda G(\lambda;\phi,\psi) = -J_{c_\psi}(\pi_{\phi,\psi,\lambda}). \tag{26}$$

The dual problem is

$$d^* = \min_{\lambda\geq 0} G(\lambda;\phi,\psi).$$

Let λ^* be an optimal solution of the dual problem. If $\lambda^* > 0$, the first-order optimality condition gives

$$\nabla_\lambda G(\lambda^*;\phi,\psi) = 0.$$

Therefore, by (26),

$$J_{c_\psi}(\pi_{\phi,\psi,\lambda^*}) = 0.$$

If $\lambda^* = 0$, the boundary optimality condition gives

$$\nabla_\lambda G(\lambda^*;\phi,\psi) \geq 0.$$

Since

$$\nabla_\lambda G(\lambda^*;\phi,\psi) = -J_{c_\psi}(\pi_{\phi,\psi,\lambda^*}),$$

we obtain

$$J_{c_\psi}(\pi_{\phi,\psi,\lambda^*}) \leq 0.$$

Thus, in both cases,

$$J_{c_\psi}(\pi_{\phi,\psi,\lambda^*}) \leq 0,$$

which means that $\pi_{\phi,\psi,\lambda^*}$ is feasible for the primal constrained RL problem. Moreover, the complementary slackness condition holds:

$$\lambda^* J_{c_\psi}(\pi_{\phi,\psi,\lambda^*}) = 0.$$

Define

$$F(\pi;\phi) = J_{r_\phi}(\pi) + \mathcal{H}(\pi).$$

Since $\pi_{\phi,\psi,\lambda^*}$ is primal feasible,

$$F(\pi_{\phi,\psi,\lambda^*};\phi) \leq p^*.$$

By weak duality,

$$p^* \leq d^*.$$

Moreover,

$$\begin{aligned} d^* &= G(\lambda^*;\phi,\psi) \\ &= F(\pi_{\phi,\psi,\lambda^*};\phi) - \lambda^* J_{c_\psi}(\pi_{\phi,\psi,\lambda^*}). \end{aligned}$$

Using complementary slackness,

$$\lambda^* J_{c_\psi}(\pi_{\phi,\psi,\lambda^*}) = 0,$$

we obtain

$$d^* = F(\pi_{\phi,\psi,\lambda^*};\phi).$$

Therefore,

$$F(\pi_{\phi,\psi,\lambda^*};\phi) \leq p^* \leq d^* = F(\pi_{\phi,\psi,\lambda^*};\phi).$$

Hence,

$$p^* = d^*$$

and

$$F(\pi_{\phi,\psi,\lambda^*};\phi) = p^*.$$

Therefore, $\pi_{\phi,\psi,\lambda^*}$ is an optimal policy for the primal constrained RL problem. □

E Hypergradient calculation

In order to simplify the computation, we use $f(\tau_0,\tau_1) = (1-y_r)\log\sigma(J_{r_\phi}(\tau_0) - J_{r_\phi}(\tau_1)) + y_r\log\sigma(J_{r_\phi}(\tau_1) - J_{r_\phi}(\tau_0)) + (1-y_c)\log\sigma(J_{c_\psi}(\tau_0) - J_{c_\psi}(\tau_1)) + y_c\log\sigma(J_{c_\psi}(\tau_1) - J_{c_\psi}(\tau_0)) + \log\sigma(s_0 J_{c_\psi}(\tau_0)) + \log\sigma(s_1 J_{c_\psi}(\tau_1))$ for the following calculation.

The hypergradient corresponding to ϕ is calculated as follows:

$$\begin{aligned} &\nabla_\phi F(\phi,\psi,\lambda^*(\phi,\psi)) \\ &= \nabla_\phi \sum_{(\tau_0,\tau_1)} f(\tau_0,\tau_1) D(\tau;\lambda^*(\phi,\psi)) \\ &= \sum_{(\tau_0,\tau_1)} \nabla_\phi f(\tau_0,\tau_1) D(\tau;\lambda^*(\phi,\psi)) + f(\tau_0,\tau_1)\nabla_\phi D(\tau;\lambda^*(\phi,\psi)) \\ &= \sum_{(\tau_0,\tau_1)} \nabla_\phi f(\tau_0,\tau_1) D(\tau;\lambda^*(\phi,\psi)) + f(\tau_0,\tau_1)\nabla_\phi \ln(D(\tau;\lambda^*(\phi,\psi)))D(\tau;\lambda^*(\phi,\psi)) \\ &= E_{D(\tau;\lambda^*(\phi,\psi))}[\nabla_\phi f(\tau_0,\tau_1) + f(\tau_0,\tau_1)\nabla_\phi(\ln(h_l(y_r,y_c,s_0,s_1|\tau_0,\tau_1)) + \ln(\rho(\tau_0;\lambda^*(\phi,\psi))) \\ &+ \ln(\rho(\tau_1;\lambda^*(\phi,\psi))))] \\ &= E_{D(\tau;\lambda^*(\phi,\psi))}[\nabla_\phi f(\tau_0,\tau_1) + f(\tau_0,\tau_1)(\nabla_\phi \ln(\rho(\tau_0;\lambda^*(\phi,\psi))) + \nabla_\phi \ln(\rho(\tau_1;\lambda^*(\phi,\psi))))] \\ &= E_{D(\tau;\lambda^*(\phi,\psi))}[\nabla_\phi f(\tau_0,\tau_1) + f(\tau_0,\tau_1)(\sum_{h=0}^{H-1} \nabla_\phi \ln(\pi_{\phi,\psi,\lambda^*(\phi,\psi)}(a_h^0|s_h^0)) \\ &+ \sum_{h=0}^{H-1} \nabla_\phi \ln(\pi_{\phi,\psi,\lambda^*(\phi,\psi)}(a_h^1|s_h^1)))], \end{aligned} \tag{27}$$

where $a_h^0, s_h^0 \in \tau_0$ and $a_h^1, s_h^1 \in \tau_1$. Analogously, we can get

$$\begin{aligned}
&\nabla_\psi F(\phi, \psi, \lambda^*(\phi, \psi)) \\
&= E_{D(\tau;\lambda^*(\phi,\psi))}[\nabla_\psi f(\tau_0, \tau_1) + f(\tau_0, \tau_1)(\sum_{h=0}^{H-1} \nabla_\psi \ln(\pi_{\phi,\psi,\lambda^*(\phi,\psi)}(a_h^0|s_h^0)) \\
&+ \sum_{h=0}^{H-1} \nabla_\psi \ln(\pi_{\phi,\psi,\lambda^*(\phi,\psi)}(a_h^1|s_h^1)))],
\end{aligned} \tag{28}$$

and

$$\begin{aligned}
&\nabla_\lambda F(\phi, \psi, \lambda^*(\phi, \psi)) \\
&= E_{D(\tau;\lambda^*(\phi,\psi))}[f(\tau_0, \tau_1)(\sum_{h=0}^{H-1} \nabla_\lambda \ln(\pi_{\phi,\psi,\lambda^*(\phi,\psi)}(a_h^0|s_h^0)) \\
&+ \sum_{h=0}^{H-1} \nabla_\lambda \ln(\pi_{\phi,\psi,\lambda^*(\phi,\psi)}(a_h^1|s_h^1)))]
\end{aligned} \tag{29}$$

Next the terms in $\nabla_\phi F(\phi, \psi, \lambda^*(\phi, \psi))$, $\nabla_\psi F(\phi, \psi, \lambda^*(\phi, \psi))$, and $\nabla_\lambda F(\phi, \psi, \lambda^*(\phi, \psi))$ are calculated as follows:

$$\begin{aligned}
&\nabla_\phi \ln(\sigma(J_{r_\phi}(\tau_0) - J_{r_\phi}(\tau_1))) \\
&= \nabla_\phi \ln(\frac{\exp(J_{r_\phi}(\tau_0))}{\exp(J_{r_\phi}(\tau_0)) + \exp(J_{r_\phi}(\tau_1))}) \\
&= \nabla_\phi J_{r_\phi}(\tau_0) - \nabla_\phi \ln(\exp(J_{r_\phi}(\tau_0)) + \exp(J_{r_\phi}(\tau_1))) \\
&= \nabla_\phi J_{r_\phi}(\tau_0) + -\frac{\nabla_\phi J_{r_\phi}(\tau_0) \exp(J_{r_\phi}(\tau_0)) + \nabla_\phi J_{r_\phi}(\tau_1) \exp(J_{r_\phi}(\tau_1))}{\exp(J_{r_\phi}(\tau_0)) + \exp(J_{r_\phi}(\tau_1))}
\end{aligned} \tag{30}$$

Analogously, we can get

$$\nabla_\phi \ln(\sigma(J_{r_\phi}(\tau_1) - J_{r_\phi}(\tau_0))) = \nabla_\phi J_{r_\phi}(\tau_1) - \frac{\nabla_\phi J_{r_\phi}(\tau_0) \exp(J_{r_\phi}(\tau_0)) + \nabla_\phi J_{r_\phi}(\tau_1) \exp(J_{r_\phi}(\tau_1))}{\exp(J_{r_\phi}(\tau_0)) + \exp(J_{r_\phi}(\tau_1))} \tag{31}$$

$$\nabla_\phi \ln(\sigma(J_{c_\psi}(\tau_0) - J_{c_\psi}(\tau_1))) = 0 \tag{32}$$

$$\nabla_\phi \ln(\sigma(J_{c_\psi}(\tau_1) - J_{c_\psi}(\tau_0))) = 0 \tag{33}$$

$$\nabla_\psi \ln(\sigma(J_{c_\psi}(\tau_0) - J_{c_\psi}(\tau_1))) = \nabla_\psi J_{c_\psi}(\tau_0) - \frac{\nabla_\psi J_{c_\psi}(\tau_0) \exp(J_{c_\psi}(\tau_0)) + \nabla_\psi J_{c_\psi}(\tau_1) \exp(J_{c_\psi}(\tau_1))}{\exp(J_{c_\psi}(\tau_0)) + \exp(J_{c_\psi}(\tau_1))} \tag{34}$$

$$\nabla_\psi \ln(\sigma(J_{c_\psi}(\tau_1) - J_{c_\psi}(\tau_0))) = \nabla_\psi J_{c_\psi}(\tau_1) - \frac{\nabla_\psi J_{c_\psi}(\tau_0) \exp(J_{c_\psi}(\tau_0)) + \nabla_\psi J_{c_\psi}(\tau_1) \exp(J_{c_\psi}(\tau_1))}{\exp(J_{c_\psi}(\tau_0)) + \exp(J_{c_\psi}(\tau_1))} \tag{35}$$

$$\nabla_\psi \ln(\sigma(J_{r_\phi}(\tau_0) - J_{r_\phi}(\tau_1))) = 0 \tag{36}$$

$$\nabla_\psi \ln(\sigma(J_{r_\phi}(\tau_1) - J_{r_\phi}(\tau_0))) = 0 \tag{37}$$

$$
\begin{aligned}
&\nabla_\psi \ln(\sigma(sJ_{c_\psi}(\tau))) \\
&= \nabla_\phi \ln(\frac{1}{1+\exp(sJ_{c_\psi}(\tau))}) \\
&= -\nabla_\phi \ln(1+\exp(sJ_{c_\psi}(\tau)) \\
&= -\frac{s\nabla_\phi J_{c_\psi}(\tau)\exp(sJ_{c_\psi}(\tau))}{1+\exp(sJ_{c_\psi}(\tau))}
\end{aligned}
\tag{38}
$$

Analogously, we can get

$$
\nabla_\phi \ln(\sigma(sJ_{c_\psi}(\tau))) = -\frac{s\nabla_\phi J_{c_\psi}(\tau)\exp(sJ_{c_\psi}(\tau))}{1+\exp(sJ_{c_\psi}(\tau))} = 0 \tag{39}
$$

Therefore, we can get

$$
\begin{aligned}
&\nabla_\phi f(\tau_0,\tau_1) \\
&= (1-y_r)\nabla_\phi \ln(\sigma(J_{r_\phi}(\tau_0)-J_{r_\phi}(\tau_1))) + y_r\nabla_\phi \ln(\sigma(J_{r_\phi}(\tau_1)-J_{r_\phi}(\tau_0))) \\
&+ (1-y_c)\nabla_\phi \ln(\sigma(J_{c_\psi}(\tau_0)-J_{c_\psi}(\tau_1))) + y_c\nabla_\phi \ln(\sigma(J_{c_\psi}(\tau_1)-J_{c_\psi}(\tau_0))) \\
&+ \nabla_\phi \ln(\sigma(s_0 J_{c_\psi}(\tau_0))) + \nabla_\phi \ln(\sigma(s_1 J_{c_\psi}(\tau_1))) \\
&= (1-y_r)\nabla_\phi J_{r_\phi}(\tau_0) + y_r\nabla_\phi J_{r_\phi}(\tau_1) - \frac{\nabla_\phi J_{r_\phi}(\tau_0)\exp(J_{r_\phi}(\tau_0)) + \nabla_\phi J_{r_\phi}(\tau_1)\exp(J_{r_\phi}(\tau_1))}{\exp(J_{r_\phi}(\tau_0))+\exp(J_{r_\phi}(\tau_1))}
\end{aligned}
\tag{40}
$$

$$
\begin{aligned}
&\nabla_\psi f(\tau_0,\tau_1) \\
&= (1-y_r)\nabla_\psi \ln(\sigma(J_{r_\phi}(\tau_0)-J_{r_\phi}(\tau_1))) + y_r\nabla_\psi \ln(\sigma(J_{r_\phi}(\tau_1)-J_{r_\phi}(\tau_0))) \\
&+ (1-y_c)\nabla_\psi \ln(\sigma(J_{c_\psi}(\tau_0)-J_{c_\psi}(\tau_1))) + y_c\nabla_\psi \ln(\sigma(J_{c_\psi}(\tau_1)-J_{c_\psi}(\tau_0))) \\
&+ \nabla_\psi \ln(\sigma(s_0 J_{c_\psi}(\tau_0))) + \nabla_\psi \ln(\sigma(s_1 J_{c_\psi}(\tau_1))) \\
&= (1-y_c)\nabla_\phi J_{c_\psi}(\tau_0) + y_c\nabla_\psi J_{c_\psi}(\tau_1) - \frac{\nabla_\psi J_{c_\psi}(\tau_0)\exp(J_{c_\psi}(\tau_0)) + \nabla_\psi J_{c_\psi}(\tau_1)\exp(J_{c_\psi}(\tau_1))}{\exp(J_{c_\psi}(\tau_0))+\exp(J_{c_\psi}(\tau_1))} \\
&- \frac{s_0\nabla_\phi J_{c_\psi}(\tau_0)\exp(s_0 J_{c_\psi}(\tau_0))}{1+\exp(s_0 J_{c_\psi}(\tau_0))} - \frac{s_1\nabla_\phi J_{c_\psi}(\tau_1)\exp(s_1 J_{c_\psi}(\tau_1))}{1+\exp(s_1 J_{c_\psi}(\tau_1))}
\end{aligned}
\tag{41}
$$

The necessary gradients and elements used for approximating subdifferetial are shown as follows:

$$
\begin{aligned}
&\nabla^2_\lambda G(\lambda;\phi,\psi) \\
&= \nabla_\lambda J_{c_\psi}(\pi_{\phi,\psi,\lambda}) \\
&= \int_{s\in S} f(\pi_{\phi,\psi,\lambda},s)\int_{a\in A}\pi_{\phi,\psi,\lambda}(a|s)\nabla_\lambda \ln(\pi_{\phi,\psi,\lambda}(a|s))J_{c_\psi}(\pi_{\phi,\psi,\lambda},s,a)dads
\end{aligned}
\tag{42}
$$

$$
\begin{aligned}
&\nabla_{\phi\lambda} G(\lambda;\phi,\psi) \\
&= \nabla_\phi J_{c_\psi}(\pi_{\phi,\psi,\lambda}) \\
&= E^{\pi_{\phi,\psi,\lambda}}[\nabla_\phi \ln(\pi_{\phi,\psi,\lambda}(a|s))J_{c_\psi}(\pi_{\phi,\psi,\lambda},s,a)]
\end{aligned}
\tag{43}
$$

$$
\begin{aligned}
&\nabla_{\psi\lambda} G(\lambda;\phi,\psi) \\
&= \nabla_\psi J_{c_\psi}(\pi_{\phi,\psi,\lambda}) \\
&= E^{\pi_{\phi,\psi,\lambda}}[\nabla_\psi \ln(\pi_{\phi,\psi,\lambda}(a|s))J_{c_\psi}(\pi_{\phi,\psi,\lambda},s,a) + \nabla_\psi c_\psi]
\end{aligned}
\tag{44}
$$

Through these gradients we can get calculate the hypergradients compute $H_\phi(\phi,\psi,\lambda^*(\phi,\psi))$ and $H_\psi(\phi,\psi,\lambda^*(\phi,\psi))$ follow equations (7) and (8).

Recall that the Lagrangian of the lower-level optimization problem is $L(\phi,\psi,\lambda,\nu) \triangleq G(\lambda;\phi,\psi)-\nu\lambda$. According to KKT stationarity, we can get $\nu(\phi^k,\psi^k) = -J_{c_{\psi^k}}(\pi_{\phi^k,\psi^k,\lambda^k})$. When $\lambda^*(\phi,\psi)=0$ and $\nu(\phi,\psi)>0$, the constraint is strictly active, corresponding to the equation (4) in [20], we can get

$$[\nabla_\phi\lambda^*(\phi,\psi)^T,\nabla_\phi\nu(\phi,\psi)^T]^T = -\begin{bmatrix}\nabla^2_\lambda L(\phi,\psi,\lambda,\nu) & -\nabla_\lambda\lambda\\ -\nabla_\lambda\lambda & 0\end{bmatrix}^{-1}[\nabla^2_{\phi\lambda}L(\phi,\psi,\lambda,\nu)^T,-\nabla_\phi\lambda^T]^T$$

$$[\nabla_\phi\lambda^*(\phi,\psi)^T,\nabla_\phi\nu(\phi,\psi)^T]^T = -\begin{bmatrix}\nabla^2_\lambda G(\lambda;\phi,\psi) & -1\\ -1 & 0\end{bmatrix}^{-1}[\nabla^2_{\phi\lambda}G(\lambda;\phi,\psi)^T,0]^T,$$

$$-[\nabla^2_{\phi\lambda}G(\lambda;\phi,\psi)^T,0]^T = \begin{bmatrix}\nabla^2_\lambda G(\lambda;\phi,\psi) & -1\\ -1 & 0\end{bmatrix}[\nabla_\phi\lambda^*(\phi,\psi)^T,\nabla_\phi\nu(\phi,\psi)^T]^T,$$

from the above equation, we can get $\nabla_\phi\lambda^*(\phi,\psi)=0$ and $\nabla_\phi\nu(\phi,\psi)=\nabla^2_{\phi\lambda}G(\lambda;\phi,\psi)$. Similarly, we can get $\nabla_\psi\lambda^*(\phi,\psi)=0$ and $\nabla_\psi\nu(\phi,\psi)=\nabla^2_{\psi\lambda}G(\lambda;\phi,\psi)$ through the same process.

When $\lambda^*(\phi,\psi)>0$ and $\nu(\phi,\psi)=0$, we can get $\nabla_\psi\nu(\phi,\psi)=0$, and $\nabla_\psi\lambda^*(\phi,\psi) = -\nabla^2_\lambda G(\lambda;\phi,\psi)^{-1}\nabla^2_{\psi\lambda}G(\lambda;\phi,\psi)$.

F Proof of Proposition 4.2

Proof. Let $L_{\lambda,\phi}$ and $L_{\nu,\phi}$ be local Lipschitz constants of $\lambda^*(\phi,\psi)$ and $\nu(\phi,\psi)$, respectively, on $\mathcal{B}(\phi',\epsilon)$. Under the smoothness and strong-convexity conditions in Appendices H and I, the KKT solution mapping (λ^*,ν) is locally Lipschitz. Hence, for every $\phi''\in\mathcal{B}(\phi',\epsilon)$,

$$\lambda^*(\phi'',\psi)\geq\lambda^*(\phi',\psi)-L_{\lambda,\phi}\epsilon,$$
$$\nu(\phi'',\psi)\geq\nu(\phi',\psi)-L_{\nu,\phi}\epsilon.$$

Therefore, the condition

$$\lambda^*(\phi',\psi)>L_{\lambda,\phi}\epsilon$$

guarantees that $\lambda^*(\phi'',\psi)>0$ throughout the neighborhood. This excludes the case $\lambda^*(\phi'',\psi)=0$ throughout the neighborhood, so $\lambda^*(\phi,\psi)$ is continuously differentiable.

Similarly, the condition

$$\nu(\phi',\psi)>L_{\nu,\phi}\epsilon$$

guarantees that $\nu(\phi'',\psi)>0$ throughout the neighborhood. Complementary slackness consequently forces $\lambda^*(\phi'',\psi)=0$ throughout the neighborhood, so $\lambda^*(\phi,\psi)$ is locally constant and therefore continuously differentiable. Thus, these sufficient conditions certify differentiability by ensuring that either

$$\lambda^*>0,\quad \nu=0,\qquad \text{or}\qquad \lambda^*=0,\quad \nu>0$$

is maintained throughout the neighborhood. The potentially nondifferentiable case $\lambda^*=0,\nu=0$ is excluded by these conditions. The argument with respect to ψ is analogous. □

In the numerical implementation, at a point where the corresponding derivatives exist, we estimate the local Lipschitz constants using a margin $\delta>0$ chosen to satisfy

$$\widehat{L}_{\lambda,\phi}\triangleq\|\nabla_\phi\lambda^*(\phi',\psi)\|+\delta\geq\sup_{\bar\phi\in\mathcal{B}(\phi',\epsilon)}\|\nabla_\phi\lambda^*(\bar\phi,\psi)\|\geq L_{\lambda,\phi},$$

and

$$\widehat{L}_{\nu,\phi}\triangleq\|\nabla_\phi\nu(\phi',\psi)\|+\delta\geq\sup_{\bar\phi\in\mathcal{B}(\phi',\epsilon)}\|\nabla_\phi\nu(\bar\phi,\psi)\|\geq L_{\nu,\phi}.$$

These bounds allow us to take $\widehat{L}_{\lambda,\phi}$ and $\widehat{L}_{\nu,\phi}$ as Lipschitz constants on $\mathcal{B}(\phi',\epsilon)$. Therefore, with δ chosen as above, Proposition 4.2 uses the following sufficient conditions to guarantee continuous differentiability throughout the neighborhood:

$$\lambda^*(\phi',\psi)>\big(\|\nabla_\phi\lambda^*(\phi',\psi)\|+\delta\big)\epsilon$$

or

$$\nu(\phi',\psi)>\big(\|\nabla_\phi\nu(\phi',\psi)\|+\delta\big)\epsilon.$$

G Proof of Lemma 4.3

Proof. Define $S'_\phi = conv\{H_\phi(\phi'', \psi, \lambda^*(\phi'', \psi))|\phi'' \in \mathcal{B}(\phi, \epsilon), \lambda^*(\phi'', \psi)$ differentiable$\}$. For all $\phi \in \mathbb{R}^{d_\phi}$, consider there exists a small $\epsilon > 0$. Through Taylor's theorem, we can get $|\|H(\phi, \psi, \lambda^*(\phi, \psi))\| - \|H(\phi', \psi, \lambda^*(\phi', \psi))\|| < \mathcal{O}(\epsilon)$ for all $\phi' \in \mathcal{B}(\phi, \epsilon)$. Therefore, we can get $|\|H(\phi, \psi, \lambda^*(\phi, \psi))\| - d(0, S'_\phi)| < \mathcal{O}(\epsilon)$.

From Proposition 4 of [20], we can conclude that $|\|H(\phi', \psi, 0)\| - d(0, S'_{\phi'})| = |\|\nabla_\phi F(\phi', \psi, 0)\| - d(0, S'_\phi)| < \mathcal{O}(\epsilon)$ for all $\phi' \in \mathcal{B}(\phi, \epsilon)$.

For any $g \in S_\phi$, we can conclude $g = \kappa H_\phi(\phi, \psi, \lambda^*(\phi, \psi)) + (1-\kappa)\nabla_\phi F(\phi, \psi, 0)$ with $\kappa \in [0, 1]$. Then we have

$$\begin{aligned}
|\|g\| - d(0, S'_{\phi'})| &\leq |\kappa\|H_\phi(\phi, \psi, \lambda^*(\phi, \psi))\| + (1-\kappa)\|\nabla_\phi F(\phi, \psi, 0)\| - d(0, S'_\phi)|| \\
&= |\kappa(\|H_\phi(\phi, \psi, \lambda^*(\phi, \psi))\| - d(0, S'_\phi)) + (1-\kappa)\|(\nabla_\phi F(\phi, \psi, 0)\| - d(0, S'_\phi))|| \\
&= \kappa\mathcal{O}(\epsilon) + (1-\kappa)\mathcal{O}(\epsilon) \\
&= \mathcal{O}(\epsilon)
\end{aligned}$$

Finally, we can conclude that $|d(0, S_\phi) - d(0, S'_\phi)| = d(S_\phi, S'_\phi) \leq \mathcal{O}(\epsilon)$.

□

H Strong convexity

We show that there exists a positive constant μ_l such that $\nabla^2_\lambda G(\lambda; \phi, \psi) \geq \mu_l$ for the lower-level problem without constraint. For any lower-level iteration number k_l, we can get

$$\begin{aligned}
&\|\lambda_{k_l+1} - \lambda^*\|^2, \\
&= \|\lambda_{k_l} - \omega_k \hat{\nabla}^2_\lambda G(\lambda_{k_l}; \phi, \psi) - \lambda^*\|^2, \\
&= \|\lambda_{k_l} - \lambda^*\|^2 + \omega^2_{k_l}\|\hat{\nabla}^2_\lambda G(\lambda_{k_l}; \phi, \psi)\|^2 - \omega_{k_l}\langle\hat{\nabla}^2_\lambda G(\lambda_{k_l}; \phi, \psi), \lambda_{k_l} - \lambda^*\rangle, \\
&\leq \|\lambda_{k_l} - \lambda^*\|^2 + \omega^2_k\|Hc_{max}\|^2 + \beta_{k_l}Hc_{max}\|\lambda_0 - \sum_{i=0}^{k_l-1}\omega_i\hat{\nabla}^2_\lambda G(\lambda_{k_l}; \phi, \psi) - \lambda^*\|, \\
&\leq \|\lambda_{k_l} - \lambda^*\|^2 + \omega^2_{k_l}\|Hc_{max}\|^2 + \omega_{k_l}Hc_{max}(\|\lambda_0 - \lambda^*\| + \|\sum_{i=0}^{k_l-1}\beta_i\hat{\nabla}^2_\lambda G(\lambda_k; \phi, \psi)\|).
\end{aligned} \tag{45}$$

Then we sum both side from $k_l = 0$ to $\bar{k}_l$, we can get

$$\begin{aligned}
&\sum_{k_l=0}^{\bar{k}_l}\|\lambda_{k_l+1} - \lambda^*\|^2, \\
&\leq \sum_{k_l=0}^{\bar{k}_l}\|\lambda_{k_l} - \lambda^*\|^2 + \beta^2_{k_l}\|Hc_{max}\|^2 + \beta_{k_l}Hc_{max}(\|\lambda_0 - \lambda^*\| + \|\sum_{i=0}^{k_l-1}\beta_i\hat{\nabla}^2_\lambda G(\lambda_k; \phi, \psi)\|), \\
&\leq \|\lambda_0 - \lambda^*\|^2 + \sum_{i=0}^{k_l-1}\beta^2_{k_l}\|Hc_{max}\|^2 + \beta_{k_l}Hc_{max}(\|\lambda_0 - \lambda^*\| + \|\sum_{i=0}^{k_l-1}\beta_i\hat{\nabla}^2_\lambda G(\lambda_k; \phi, \psi)\|).
\end{aligned} \tag{46}$$

As $\sum_{i=0}^{k_l-1}\omega^2_{k_l}$ and $\sum_{i=0}^{k_l-1}\sum_{i=0}^{k_l-1}\omega_i$ are finite sums, they are bounded. Since the lower-level problem is strictly convex, the value λ^* exists, and $\|\lambda_0 - \lambda^*\|$ is bounded. Then, all elements on the right-hand side are bounded. Therefore, $\|\lambda_{\bar{k}_l} - \lambda^*\|^2$ is bounded for any finite $\bar{k}_l \geq 0$. According to the Lemma 9 in [22], there exist a positive constant μ_l such that $\nabla^2_\lambda G(\lambda; \phi, \psi) \geq \mu_l$ without considering the constraint. As $\lambda \geq 0$ is a convex subset of $\mathbb{R}$, we can conclude that $\nabla^2_\lambda G(\lambda; \phi, \psi) \geq \mu_l$ when λ is differentiable. As a result, $\nabla^2_\lambda G(\lambda; \phi, \psi) \geq \mu_l$ is invertible and the hypergradient exists.

I Lipschitz continuous and boundness

Without the constraint, we have the following lemmas.

Lemma I.1. *Without the constraint, the functions $F(\phi,\psi,\lambda)$ and $G(\lambda;\phi,\psi)$ have the following properties.*
1. For any $(\phi,\psi,\lambda)\in\mathbb{R}^{d_\phi}\times\mathbb{R}^{d_\psi}\times\mathbb{R}$, the gradients $\nabla_\phi F(\phi,\psi,\lambda)$, $\nabla_\psi F(\phi,\psi,\lambda)$ and $\nabla_\lambda F(\phi,\psi,\lambda)$ are Lipschitz continuous with respect to λ with constants $L_{F_\phi}>0$, $L_{F_\psi}>0$ and $L_{F_\lambda}>0$, respectively. The gradient $\nabla_\lambda F(\phi,\psi,\lambda)$ is Lipschitz continuous with respect to ϕ and ψ with constants $L'_{F_\lambda}>0$ and $L''_{F_\lambda}>0$. The gradients $\nabla_\phi F(\phi,\psi,\lambda)$ is Lipschitz continuous with respect to ϕ with constant L'_{F_ϕ}. The $\nabla_\psi F(\phi,\psi,\lambda)$ is Lipschitz continuous with respect to ψ with constant L''_{F_ψ}.
2. For any $(\phi,\psi,\lambda)\in\mathbb{R}^{d_\phi}\times\mathbb{R}^{d_\psi}\times\mathbb{R}$, we have $\|\nabla_\lambda F(\phi,\psi,\lambda)\|\le C_{F_\lambda}$ with $C_{F_\lambda}>0$
3. For any $(\phi,\psi,\lambda)\in\mathbb{R}^{d_\phi}\times\mathbb{R}^{d_\psi}\times\mathbb{R}$, the gradients $\nabla_\lambda G(\lambda;\phi,\psi)$, $\nabla^2_\lambda G(\lambda;\phi,\psi)$, $\nabla^2_{\phi\lambda} G(\lambda;\phi,\psi)$ and $\nabla^2_{\psi\lambda} G(\lambda;\phi,\psi)$ are Lipschitz continuous with respect to λ with constants $L_{G_\lambda}>0$, $L_{G_{\lambda\lambda}}>0$, $L_{G_{\phi\lambda}}>0$ and $L_{G_{\psi\lambda}}$, respectively. The gradients $\nabla^2_\lambda G(\lambda;\phi,\psi)$, $\nabla^2_{\phi\lambda} G(\lambda;\phi,\psi)$ and $\nabla^2_{\psi\lambda} G(\lambda;\phi,\psi)$ are Lipschitz continuous with respect to ϕ with constants $L'_{G_{\lambda\lambda}}>0$, $L'_{G_{\phi\lambda}}>0$ and $L'_{G_{\psi\lambda}}>0$, respectively. The gradients $\nabla^2_\lambda G(\lambda;\phi,\psi)$, $\nabla^2_{\phi\lambda} G(\lambda;\phi,\psi)$ and $\nabla^2_{\psi\lambda} G(\lambda;\phi,\psi)$ are Lipschitz continuous with respect to ψ with constants $L''_{G_{\lambda\lambda}}>0$, $L''_{G_{\phi\lambda}}>0$ and $L''_{G_{\psi\lambda}}>0$, respectively.
4. For any $(\phi,\psi,\lambda)\in\mathbb{R}^{d_\phi}\times\mathbb{R}^{d_\psi}\times\mathbb{R}$, we have $\|\nabla^2_{\phi\lambda} G(\phi,\psi,\lambda)\|\le C_{G_{\phi\lambda}}$ and $\|\nabla^2_{\psi\lambda} G(\phi,\psi,\lambda)\|\le C_{G_{\psi\lambda}}$ with $C_{G_{\phi\lambda}}>0$ and $C_{G_{\psi\lambda}}>0$.

First, we show the lipschitz continuous of $H_\phi(\phi,\psi,\lambda)$ with respect to ϕ and λ. For any $(\phi,\psi)\in\mathbb{R}^{d_\phi}\times\mathbb{R}^{d_\psi}$, $\lambda_1\in\mathbb{R}$ and $\lambda_2\in\mathbb{R}$, we can get

$$
\begin{aligned}
&\|H(\phi,\psi,\lambda_1)-H(\phi,\psi,\lambda_2)\|\\
&=\|\nabla_\phi F(\phi,\psi,\lambda_1)+\nabla^2_{\phi\lambda}G(\lambda_1;\phi,\psi)[\nabla^2_\lambda G(\lambda_1;\phi,\psi)]^{-1}\nabla_\lambda F(\phi,\psi,\lambda_1)\\
&-\nabla_\phi F(\phi,\psi,\lambda_2)+\nabla^2_{\phi\lambda}G(\lambda_2;\phi,\psi)[\nabla^2_\lambda G(\lambda_2;\phi,\psi)]^{-1}\nabla_\lambda F(\phi,\psi,\lambda_2)\|\\
&\le\|\nabla_\phi F(\phi,\psi,\lambda_1)-\nabla_\phi F(\phi,\psi,\lambda_2)\|\\
&+\|\nabla^2_{\phi\lambda}G(\lambda_1;\phi,\psi)[\nabla^2_\lambda G(\lambda_1;\phi,\psi)]^{-1}(\nabla_\lambda F(\phi,\psi,\lambda_1)-\nabla_\lambda F(\phi,\psi,\lambda_2))\|\\
&+\|(\nabla^2_{\phi\lambda}G(\lambda_1;\phi,\psi)-\nabla^2_{\phi\lambda}G(\lambda_2;\phi,\psi))[\nabla^2_\lambda G(\lambda_1;\phi,\psi)]^{-1}\nabla_\lambda F(\phi,\psi,\lambda_2)\|\\
&+\|\nabla^2_{\phi\lambda}G(\lambda_2;\phi,\psi)([\nabla^2_\lambda G(\lambda_1;\phi,\psi)]^{-1}-[\nabla^2_\lambda G(\lambda_2;\phi,\psi)]^{-1})\nabla_\lambda F(\phi,\psi,\lambda_2)\|
\end{aligned}
$$

For any two invertible matrix X_1 and X_2, we can get $\|X_1^{-1}-X_2^{-1}\|=\|X_1^{-1}(X_1-X_2)X_2^{-1}\le\|X_1^{-1}\|\|X_1-X_2\|\|X_2^{-1}\|$. Therefore, keep solving,

$$
\begin{aligned}
&\|H(\phi,\psi,\lambda_1)-H(\phi,\psi,\lambda_2)\|\\
&\le L_{F_\phi}\|\lambda_1-\lambda_2\|+\frac{L_{F_\lambda}C_{G_{\phi\lambda}}}{\mu_l}\|\lambda_1-\lambda_2\|+\frac{C_{F_\lambda}L_{G_{\phi\lambda}}}{\mu_l}\|\lambda_1-\lambda_2\|+\frac{C_{F_\lambda}L_{G_{\lambda\lambda}}C_{G_{\phi\lambda}}}{\mu_l^2}\|\lambda_1-\lambda_2\|\\
&=(L_{F_\phi}+\frac{L_{F_\lambda}C_{G_{\phi\lambda}}}{\mu_l}+\frac{C_{F_\lambda}L_{G_{\lambda\lambda}}C_{G_{\phi\lambda}}}{\mu_l^2}+\frac{C_{F_\lambda}L_{G_{\lambda\lambda}}C_{G_{\phi\lambda}}}{\mu_l^2})\|\lambda_1-\lambda_2\|
\end{aligned}
$$

For any $(\psi, \lambda) \in \mathbb{R}^{d_\psi} \times \mathbb{R}$, $\phi_1 \in \mathbb{R}^{d_\phi}$ and $\phi_2 \in \mathbb{R}^{d_\phi}$, we can get

$$
\begin{aligned}
&\|H(\phi_1, \psi, \lambda_1) - H(\phi_2, \psi, \lambda)\| \\
&= \|\nabla_\phi F(\phi_1, \psi, \lambda_1) + \nabla^2_{\phi\lambda} G(\lambda; \phi_1, \psi)[\nabla^2_\lambda G(\lambda; \phi_1, \psi)]^{-1} \nabla_\lambda F(\phi_1, \psi, \lambda) \\
&- \nabla_\phi F(\phi_2, \psi, \lambda_1) + \nabla^2_{\phi\lambda} G(\lambda; \phi_2, \psi)[\nabla^2_\lambda G(\lambda; \phi_2, \psi)]^{-1} \nabla_\lambda F(\phi_2, \psi, \lambda)\| \\
&\leq \|\nabla_\phi F(\phi_1, \psi, \lambda_1) - \nabla_\phi F(\phi_2, \psi, \lambda_1)\| \\
&+ \|\nabla^2_{\phi\lambda} G(\lambda; \phi_1, \psi)[\nabla^2_\lambda G(\lambda; \phi_1, \psi)]^{-1} (\nabla_\lambda F(\phi_1, \psi, \lambda) - \nabla_\lambda F(\phi_2, \psi, \lambda))\| \\
&+ \|(\nabla^2_{\phi\lambda} G(\lambda; \phi_1, \psi) - \nabla^2_{\phi\lambda} G(\lambda; \phi_2, \psi))[\nabla^2_\lambda G(\lambda; \phi_1, \psi)]^{-1} \nabla_\lambda F(\phi_2, \psi, \lambda)\| \\
&+ \|\nabla^2_{\phi\lambda} G(\lambda; \phi_2, \psi)([\nabla^2_\lambda G(\lambda; \phi_1, \psi)]^{-1} - [\nabla^2_\lambda G(\lambda; \phi_2, \psi)]^{-1}) \nabla_\lambda F(\phi_2, \psi, \lambda)\| \\
&\leq L'_{F_\phi} \|\phi_1 - \phi_2\| + \frac{L'_{F_\lambda} C_{G_{\phi\lambda}}}{\mu_l} \|\phi_1 - \phi_2\| + \frac{C_{F_\lambda} L'_{G_{\phi\lambda}}}{\mu_l} \|\phi_1 - \phi_2\| + \frac{C_{F_\lambda} L'_{G_{\lambda\lambda}} C_{G_{\phi\lambda}}}{\mu_l^2} \|\phi_1 - \phi_2\| \\
&= (L'_{F_\phi} + \frac{L'_{F_\lambda} C_{G_{\phi\lambda}}}{\mu_l} + \frac{C_{F_\lambda} L'_{G_{\phi\lambda}}}{\mu_l} + \frac{C_{F_\lambda} L'_{G_{\lambda\lambda}} C_{G_{\phi\lambda}}}{\mu_l^2}) \|\phi_1 - \phi_2\|
\end{aligned}
$$

Let$L_\lambda = L_{F_\phi} + \frac{L_{F_\lambda} C_{G_{\phi\lambda}}}{\mu_l} + \frac{C_{F_\lambda} L_{G_{\lambda\lambda}} C_{G_{\phi\lambda}}}{\mu_l^2} + \frac{C_{F_\lambda} L_{G_{\lambda\lambda}} C_{G_{\phi\lambda}}}{\mu_l^2}$ and $L_\phi = L'_{F_\phi} + \frac{L'_{F_\lambda} C_{G_{\phi\lambda}}}{\mu_l} + \frac{C_{F_\lambda} L'_{G_{\phi\lambda}}}{\mu_l} + \frac{C_{F_\lambda} L'_{G_{\lambda\lambda}} C_{G_{\phi\lambda}}}{\mu_l^2}$, we have $\|H(\phi, \psi, \lambda_1) - H(\phi, \psi, \lambda_2)\| \leq L_\lambda \|\lambda_1 - \lambda_2\|$ and $\|H(\phi_1, \psi, \lambda_1) - H(\phi_2, \psi, \lambda)\| \leq L_\phi \|\phi_1 - \phi_2\|$. Next, we have

$$
\begin{aligned}
&\|H(\phi_1, \psi, \lambda^*(\phi_1, \psi))) - H(\phi_2, \psi, \lambda^*(\phi_2, \psi))\| \\
&\leq \|H(\phi_1, \psi, \lambda^*(\phi_1, \psi))) - H(\phi_1, \psi, \lambda^*(\phi_2, \psi))\| + \|H(\phi_1, \psi, \lambda^*(\phi_2, \psi))) - H(\phi_2, \psi, \lambda^*(\phi_2, \psi))\| \\
&\leq L_\lambda \|\lambda^*(\phi_1, \psi) - \lambda^*(\phi_2, \psi)\| + L_\phi \|\phi_1 - \phi_2\| \\
&\leq (\frac{L_\lambda C_{G_{\phi\lambda}}}{\mu_g} + L_\phi) \|\phi_1 - \phi_2\|
\end{aligned}
$$

Let $L = \frac{L_\lambda C_{G_{\phi\lambda}}}{\mu_g} + L_\phi$, we can conclude that $\|H(\phi_1, \psi, \lambda^*(\phi_1, \psi))) - H(\phi_2, \psi, \lambda^*(\phi_2, \psi))\| \leq L\|\phi_1 - \phi_2\|$.

J Proof of Theorem 5.3

To unify the notation for the differentiable and non-differentiable cases, we define the theoretical first-order direction H' as follows:

$$
H'(\phi_k, \psi, \lambda^*(\phi_k, \psi)) \triangleq \begin{cases} H(\phi_k, \psi, \lambda^*(\phi_k, \psi)), & \text{if differentiability is certified by Proposition 4.2,} \\ \arg\min_{g \in \mathcal{S}_{\phi,k}} \|g\|, & \text{otherwise.} \end{cases}
$$

Thus, H' denotes the exact hypergradient in the differentiable and non-differentiable cases. The prime is a label for this theoretical direction, not a derivative of H. Here, H denotes the exact hypergradient with respect to ϕ.

We use g_ϕ^k to denote the direction computed by Algorithm 1 for the update of ϕ.

Following the guidance of [43], we choose an adaptive number of inner-loop iterations $t_k = \lceil \frac{\sqrt[4]{k+1}}{2} \rceil$. This value $\lceil \frac{\sqrt[4]{k+1}}{2} \rceil$ balances the trade-off between computational cost (minimizing inner loop iteration number) and approximation accuracy (reducing the error sufficiently fast), enabling efficient and stable convergence of the bi-level algorithm.

First, we need to quantify all approximation errors. The overall approximation error comes from two aspects, one from the approximation of the subgradient and another one is the difference between the calculated λ and λ^* from the lower-level. The approximation of the subgradient is $\mathcal{O}(\epsilon)$ as shown in Lemma 4.3. Following the idea of gradient descent, we can show that $\|\lambda_{k_l} - \lambda^*\| \leq (1 - \frac{L_{G_\lambda}}{\mu_g})^{k_l} \|\lambda_0 - \lambda^*\|$. Therefore, we conclude the approximation error at iteration k as $E_k = C_s \epsilon + (1 - \frac{L_{G_\lambda}}{\mu_g})^{k_l} \|\lambda_0 - \lambda^*\|$,where C_s is coefficient of the approximation error of the

subgradient. This discrepancy arises from the approximation of the subdifferentials and the error from solving the lower-level problem approximately.

Next, we can show

$$
\begin{aligned}
&F(\phi_{k+1}, \psi, \lambda^*(\phi_{k+1}, \psi)) \\
&\leq F(\phi_k, \psi, \lambda^*(\phi_k, \psi)) + \langle H'(\phi_k, \psi, \lambda^*(\phi_k, \psi)), \phi_{k+1} - \phi_k \rangle + \frac{L}{2}\|\phi_{k+1} - \phi_k\|^2 \\
&= F(\phi_k, \psi, \lambda^*(\phi_k, \psi)) - \alpha_k \langle H'(\phi_k, \psi, \lambda^*(\phi_k, \psi)), g_\phi^k \rangle + \frac{L\alpha_k^2}{2}\|g_\phi^k\|^2
\end{aligned}
$$

Define $g_\phi^k = H'(\phi_k, \psi, \lambda^*(\phi_k, \psi)) + E_k$. Then the expectation of $F(\phi_{k+1}, \psi, \lambda^*(\phi_{k+1}, \psi))$ becomes:

$$
\begin{aligned}
&E[F(\phi_{k+1}, \psi, \lambda^*(\phi_{k+1}, \psi))] \\
&\leq F(\phi_k, \psi, \lambda^*(\phi_k, \psi)) - \alpha_k \langle H'(\phi_k, \psi, \lambda^*(\phi_k, \psi)), H'(\phi_k, \psi, \lambda^*(\phi_k, \psi)) + E_k \rangle \\
&+ \frac{L\alpha_k^2}{2}\|H'(\phi_k, \psi, \lambda^*(\phi_k, \psi)) + g_\phi^k - H'(\phi_k, \psi, \lambda^*(\phi_k, \psi)\|^2 \\
&\leq F(\phi_k, \psi, \lambda^*(\phi_k, \psi)) - \alpha_k \langle H'(\phi_k, \psi, \lambda^*(\phi_k, \psi)), H'(\phi_k, \psi, \lambda^*(\phi_k, \psi)) + E_k \rangle \\
&+ \frac{L\alpha_k^2}{2} Var(g_\phi^k) + \frac{L\alpha_k^2}{2}\|H'(\phi_k, \psi, \lambda^*(\phi_k, \psi))\|^2 + L\alpha_k^2 \langle H'(\phi_k, \psi, \lambda^*(\phi_k, \psi)), E_k \rangle \\
&= F(\phi_k, \psi, \lambda^*(\phi_k, \psi)) - (\alpha_k - \frac{L\alpha_k^2}{2})\|H'(\phi_k, \psi, \lambda^*(\phi_k, \psi))\|^2 \\
&- (\alpha_k - L\alpha_k^2)\langle H'(\phi_k, \psi, \lambda^*(\phi_k, \psi)), E_k \rangle + \frac{L\alpha_k^2}{2} Var(g_\phi^k) + \frac{L\alpha_k^2}{2}\|E_k\|
\end{aligned}
$$

By using the fact that $2\langle H'(\phi_k, \psi, \lambda^*(\phi_k, \psi)), E_k \rangle \leq \|H'(\phi_k, \psi, \lambda^*(\phi_k, \psi))\|^2 + \|E_k\|^2$ and choosing $\alpha_k < \frac{1}{L}$, we can get

$$
\begin{aligned}
&E[F(\phi_{k+1}, \psi, \lambda^*(\phi_{k+1}, \psi))] \\
&\leq F(\phi_k, \psi, \lambda^*(\phi_k, \psi)) - \frac{\alpha_k}{2}(\|H'(\phi_k, \psi, \lambda^*(\phi_k, \psi))\|^2 - \|E_k\|^2) + \frac{L\alpha_k^2}{2} Var(g_\phi^k)
\end{aligned}
$$

Then consider from $k = 0$, we can get

$$
\begin{aligned}
&\sum_{k=0}^{K-1} \frac{\alpha_k}{2} E[\|H'(\phi_k, \psi, \lambda^*(\phi_k, \psi))\|^2] \\
&\leq F(\phi_0, \psi, \lambda^*(\phi_0, \psi)) - F^* + \sum_{k=0}^{K-1} \frac{\alpha_k}{2}\|E_k\|^2 + \frac{L\alpha_k^2}{2} Var(g_\phi^k)
\end{aligned}
$$

Then we can conclude that

$$
\begin{aligned}
&\frac{1}{K}\sum_{k=0}^{K-1} E[\|H'(\phi_k, \psi, \lambda^*(\phi_k, \psi))\|^2] \\
&\leq \frac{2}{K\alpha_k}(F(\phi_0, \psi, \lambda^*(\phi_0, \psi)) - F^*) + \frac{1}{K}\sum_{k=0}^{K-1}\|E_k\|^2 + L\alpha_k Var(g_\phi^k) \\
&\leq \frac{2}{K\alpha_k}(F(\phi_0, \psi, \lambda^*(\phi_0, \psi)) - F^*) + \frac{1}{K}\sum_{k=0}^{K-1} C_s^2\epsilon^2 + 2C_s\epsilon(1 - \frac{L_{G_\lambda}}{\mu_g})^{k_l}\|\lambda_0 - \lambda^*\| + ((1 - \frac{L_{G_\lambda}}{\mu_g})^{k_l}\|\lambda_0 - \lambda^*\|)^2 \\
&+ L\alpha_k((C_{F_\phi} + \frac{C_{G_{\phi\lambda}}C_F}{\mu_g})^2 + 2C_s\epsilon(C_{F_\phi} + \frac{C_{G_{\phi\lambda}}C_F}{\mu_g}) + C_s^2\epsilon^2)
\end{aligned}
$$

By choosing $\alpha_k = \frac{1}{L\sqrt{K}}$, $k_l = k$ and $\epsilon = \frac{1}{k^2}$, we can get

$$\begin{aligned}
&\frac{1}{K}\sum_{k=0}^{K-1} E[\|H'(\phi_k, \psi, \lambda^*(\phi_k, \psi))\|^2] \\
&\leq \frac{2L}{\sqrt{K}}(F(\phi_0, \psi, \lambda^*(\phi_0, \psi)) - F^*) + \frac{C_s^2\pi^4}{90K} + \frac{2C_s(\mu_g - L_{G_\lambda})\|\lambda_0 - \lambda^*\|\pi^2}{6LK} + \frac{(\mu_g - L_{G_\lambda})\|\lambda_0 - \lambda^*\|^2}{LK} \\
&+ \frac{(C_{F_\phi} + \frac{C_{G_{\phi\lambda}}C_F}{\mu_g})^2}{\sqrt{K}} + \frac{2C_s\pi^2(C_{F_\phi} + \frac{C_{G_{\phi\lambda}}C_F}{\mu_g})}{6\sqrt{K}} + \frac{c_s^2\pi^4}{90\sqrt{K}} \\
&= \frac{C_1}{\sqrt{K}} + \frac{C_2}{K}
\end{aligned}$$

where $C_1 = 2L(F(\phi_0, \psi, \lambda^*(\phi_0, \psi)) - F^*) + (C_{F_\phi} + \frac{C_{G_{\phi\lambda}}C_F}{\mu_g})^2 + \frac{2C_s\pi^2(C_{F_\phi} + \frac{C_{G_{\phi\lambda}}C_F}{\mu_g})}{6} + \frac{c_s^2\pi^4}{90}$ and $C_2 = \frac{C_s^2\pi^4}{90} + \frac{2C_s(\mu_g - L_{G_\lambda})\|\lambda_0 - \lambda^*\|\pi^2}{6L} + \frac{(\mu_g - L_{G_\lambda})\|\lambda_0 - \lambda^*\|^2}{L}$. Then we can conclude that

$$\min\{\|g_\phi\|^2 | g_\phi \in S_\phi\} \leq \frac{1}{K}\sum_{k=0}^{K-1} E[\|H'(\phi_k, \psi, \lambda^*(\phi_k, \psi))\|^2] \leq \mathcal{O}(\frac{1}{\sqrt{K}}) \tag{47}$$

K Proof of Theorem 5.4

Assumption K.1. The expert preference models $h_{r_e}(\cdot|\tau_0, \tau_1) \propto \exp J_{r_e}(\tau_p)$ and $h_{c_e}(\cdot|\tau_0, \tau_1) \propto \exp J_{c_e}(\tau_p)$ where τ_p is the preferred trajectory between τ_0 and τ_1.

Proof. According to the assumption K.1, for each trajectory pair $\Gamma = \{\tau_0, \tau_1\}$, the expert preference is generated according to the expert preference model $h_{r_e}(\tau|\Gamma)$:

$$h_{r_e}(\tau|\Gamma) = \frac{\exp\left(J_{r_e}(\tau)\right)}{\sum_{\tau'\in\Gamma}\exp\left(J_{r_e}(\tau')\right)}$$

where r_e is the reward function of the expert.

Define a preference model class $\mathcal{H}(\Gamma)$ defined on Γ, which is parametrized by the function r_ϕ:

$$\mathcal{H}(\Gamma) = \{h_{r_\phi}(\tau|\Gamma) = \frac{\exp\left(J_{r_\phi}(\tau)\right)}{\sum_{\tau'\in\Gamma}\exp\left(J_{r_\phi}(\tau')\right)} \quad \text{for some} \quad r_\phi \in \mathcal{F}\}.$$

Define $N_{[]}(\mathcal{H}(\Gamma), \|\cdot\|_\infty, \frac{1}{Kn})$ as the bracketing number of $\Pi(\Gamma)$.

From the upper-level objective function, we can get

$$\begin{aligned}
\frac{1}{Kn}\sum_{i=1}^{Kn}\ln(\frac{\exp\left(J_{r_\phi}(\tau)\right)}{\sum_{\tau'\in\Gamma}\exp\left(J_{r_\phi}(\tau')\right)}) &\geq \frac{1}{Kn}\sum_{i=1}^{Kn}\ln(\frac{\exp\left(J_{r_e}(\tau)\right)}{\sum_{\tau'\in\Gamma}\exp\left(J_{r_e}(\tau')\right)}) \\
\frac{1}{Kn}\sum_{i=1}^{Kn}\ln(h_{r_\phi}(\tau|\Gamma)) &\geq \frac{1}{Kn}\sum_{i=1}^{Kn}\ln(h_{r_e}(\tau|\Gamma)) \\
\frac{1}{Kn}\sum_{i=1}^{Kn}\ln(h_{r_\phi}(\tau|\Gamma)) - \frac{1}{Kn}\sum_{i=1}^{kn}\ln(h_{r_e}(\tau|\Gamma)) &\geq 0 \\
\frac{1}{Kn}\sum_{i=1}^{Kn}\ln(\frac{h_{r_\phi}(\tau|\Gamma)}{h_{r_e}(\tau|\Gamma)}) &\geq 0.
\end{aligned}$$

For any $\hat{h}$ belongs to the same bracket with h_{r_e}, according to Markov's inequality, we can get

$$P(\exp(\sum_{i=1}^{n}\frac{1}{2}\ln(\frac{\hat{h}}{h_{r_e}})) \geq \exp(\epsilon)) \leq \frac{E_\Gamma[\exp(\sum_{i=1}^{Kn}\frac{1}{2}\ln(\frac{\hat{h}}{h_{r_e}}))]}{\exp(\epsilon)}.$$

Through Boole's inequality, we can get

$$P(\forall \hat{h} \in \Pi, \exp(\sum_{i=1}^{Kn} \frac{1}{2} \ln(\frac{\hat{h}}{h_{r_e}})) \geq \exp(\epsilon)) \leq \frac{E_\Gamma[\exp(\sum_{i=1}^{Kn} \frac{1}{2} \ln(\frac{\hat{h}}{h_{r_e}}))] \cdot N_{[]}(\mathcal{H}(\Gamma), \|\cdot\|_\infty, \frac{1}{Kn})}{\exp(\epsilon)}.$$

Therefore, with probability at least δ, we can get

$$\begin{aligned}
\delta &= \frac{E_\Gamma[\exp(\sum_{i=1}^{Kn} \frac{1}{2} \ln(\frac{\hat{h}}{h_{r_e}}))] \cdot N_{[]}(\mathcal{H}(\Gamma), \|\cdot\|_\infty, \frac{1}{Kn})}{\exp(\epsilon)}, \\
\epsilon &= \ln(\frac{E_\Gamma[\exp(\sum_{i=1}^{Kn} \frac{1}{2} \ln(\frac{\hat{h}}{h_{r_e}}))] \cdot N_{[]}(\mathcal{H}(\Gamma), \|\cdot\|_\infty, \frac{1}{Kn})}{\delta} \\
\epsilon &= Kn \ln(E_\Gamma[\exp(\frac{1}{2} \ln(\frac{\hat{h}}{h_{r_e}}))]) + \ln(\frac{N_{[]}(\mathcal{H}(\Gamma), \|\cdot\|_\infty, \frac{1}{Kn})}{\delta}).
\end{aligned}$$

With probability at least $1-\delta$ and any policy π we can get

$$\begin{aligned}
\sum_{i=1}^{Kn} \frac{1}{2} \ln(\frac{\hat{h}}{h_{r_e}}) &\leq Kn \ln(E_\Gamma[\exp(\frac{1}{2} \ln(\frac{\hat{h}}{h_{r_e}}))]) + \ln(\frac{N_{[]}(\mathcal{H}(\Gamma), \|\cdot\|_\infty, \frac{1}{Kn})}{\delta}) \\
0 &\leq Kn \ln(E_\Gamma[\exp(\frac{1}{2} \ln(\frac{\hat{h}}{h_{r_e}}))]) + \ln(\frac{N_{[]}(\mathcal{H}(\Gamma), \|\cdot\|_\infty, \frac{1}{Kn})}{\delta}) \\
&\leq Kn \ln(E_\Gamma[\sqrt{\frac{\hat{h}}{h_{r_e}}}]) + \ln(\frac{N_{[]}(\mathcal{H}(\Gamma), \|\cdot\|_\infty, \frac{1}{Kn})}{\delta}) \\
&= Kn \ln(E_\Gamma[\sqrt{\hat{h} h_{r_e}}]) + \ln(\frac{N_{[]}(\mathcal{H}(\Gamma), \|\cdot\|_\infty, \frac{1}{Kn})}{\delta}).
\end{aligned}$$

Through $\ln(x) \leq x - 1$, we can get

$$1 - E_\Gamma[\sqrt{\hat{h} h_{r_e}}] \leq \frac{1}{Kn} \ln(\frac{N_{[]}(\mathcal{H}(\Gamma), \|\cdot\|_\infty, \frac{1}{Kn})}{\delta}).$$

Next, we can bound the difference between $\hat{h}$ and h_{r_e}.

$$\begin{aligned}
E_\Gamma[\|\hat{h} - h_{r_e}\|_1^2] &= E_\Gamma[\|(\hat{h}^{\frac{1}{2}} - h_{r_e}^{\frac{1}{2}})(\hat{h}^{\frac{1}{2}} + h_{r_e}^{\frac{1}{2}})\|_1^2] \\
&\leq E_\Gamma[\|\hat{h}^{\frac{1}{2}} - h_{r_e}^{\frac{1}{2}}\|_1^2] E_\Gamma[\|\hat{h}^{\frac{1}{2}} + h_{r_e}^{\frac{1}{2}}\|_1^2] \\
&\leq (2(1 - E_\Gamma[\sqrt{\hat{h} h_{r_e}}) + \frac{1}{n})(2E_\Gamma[\hat{h} + h_{r_e}]) \\
&\leq (\frac{2}{Kn} \ln(\frac{N_{[]}(\mathcal{H}(\Gamma), \|\cdot\|_\infty, \frac{1}{Kn})}{\delta}) + \frac{1}{Kn})(\frac{2}{Kn} + 4) \\
&\leq \mathcal{O}(\frac{1}{Kn} \ln(\frac{N_{[]}(\mathcal{H}(\Gamma), \|\cdot\|_\infty, \frac{1}{Kn})}{\delta}))
\end{aligned}$$

Then we can bound the difference between h_{r_ϕ} and h_{r_e}.

$$\begin{aligned}
E_\Gamma[\|h_{r_\phi} - h_{r_e}\|_1^2] &= E_\Gamma[\|h_{r_\phi} - \hat{h} + \hat{h} - h_{r_e}\|_1^2] \\
&\leq E_\Gamma[\|h_{r_\phi} - \hat{h}\|_1^2] + E_\Gamma[\|\hat{h} - h_{r_e}\|_1^2] \\
&\leq \mathcal{O}(\frac{1}{Kn} \ln(\frac{N_{[]}(\mathcal{H}(\Gamma), \|\cdot\|_\infty, \frac{1}{Kn})}{\delta}))
\end{aligned}$$

$$\begin{aligned}
\|\hat{h} - h_{r_e}\|_1^2 - \|h_{r_\phi} - h_{r_e}\|_1^2 &\leq (|\hat{h} - h_{r_e}| + |h_{r_\phi} - h_{r_e}|)(|\hat{h} - h_{r_e}| - |h_{r_\phi} - h_{r_e}|) \\
&\leq (4 + \frac{1}{Kn}) \frac{1}{Kn}.
\end{aligned}$$

We can conclude that

$$E_\Gamma[\|h_{r_\phi} - h_{r_e}\|_1^2] \leq \mathcal{O}(\frac{1}{Kn} \ln(\frac{N_{[]}(\mathcal{H}(\Gamma), \|\cdot\|_\infty, \frac{1}{Kn})}{\delta}))$$

Since the softmax function is continuous, there exists a constant $L_h \geq 0$ such that

$$|h_{r_\phi} - h_{r_e}| \leq L_h \|r_\phi - r_e\|_\infty.$$

Then, according to Lemma 2.14 in [44], we can get $N_{[]}(\mathcal{H}(\Gamma), \|\cdot\|_\infty, \frac{1}{Kn}) \leq N(\mathcal{F}, \|\cdot\|_\infty, \frac{1}{4Kn})$ for all K iterations. Overall, we can get

$$E_\Gamma[\|h_{r_\phi} - h_{r_e}\|_1^2] \leq \mathcal{O}(\frac{1}{Kn} \ln(\frac{N(\mathcal{F}, \|\cdot\|_\infty, \frac{1}{Kn})}{\delta})).$$

Next, we convert preference-model error into pairwise return-difference error. For a pair $\Gamma = \{\tau_0, \tau_1\}$, define

$$\Delta_{r_\phi}(\Gamma) \triangleq J_{r_\phi}(\tau_0) - J_{r_\phi}(\tau_1), \qquad \Delta_{r_e}(\Gamma) \triangleq J_{r_e}(\tau_0) - J_{r_e}(\tau_1).$$

For the Bradley-Terry model,

$$h_{r_\phi}(\tau_0 \mid \Gamma) = \sigma(\Delta_{r_\phi}(\Gamma)), \qquad h_{r_e}(\tau_0 \mid \Gamma) = \sigma(\Delta_{r_e}(\Gamma)),$$

where $\sigma(x) = 1/(1 + \exp(-x))$.

Following Assumption 5.1 and 5.2, we can conclude $\Delta_r(\Gamma)$ is bounded. Hence there exists $\eta > 0$ such that

$$\eta \leq h_r(\tau_0 \mid \Gamma) \leq 1 - \eta$$

for all $r \in \mathcal{F}$ and all Γ. Therefore, the inverse-logit map $\sigma^{-1}(p) = \log(p/(1-p))$ is Lipschitz on $[\eta, 1-\eta]$. Thus, there exists $C_\eta > 0$ such that

$$|\Delta_{r_{\phi^K}}(\Gamma) - \Delta_{r_e}(\Gamma)| \leq C_\eta \|h_{r_{\phi^K}}(\cdot \mid \Gamma) - h_{r_e}(\cdot \mid \Gamma)\|_1.$$

Now we bound the reward suboptimality. Let $\tau_e \sim \pi_e$ and $\tau_K \sim \pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)}$.

$$\begin{aligned}\mathrm{SubOptR}(\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)}) &= J_{r_e}(\pi_e) - J_{r_e}(\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)}) \\ &= \mathbb{E}_{\tau_e, \tau_K}\left[J_{r_e}(\tau_e) - J_{r_e}(\tau_K)\right].\end{aligned}$$

Adding and subtracting the learned reward difference gives

$$\begin{aligned}\mathrm{SubOptR}(\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)}) = \mathbb{E}_{\tau_e, \tau_K}&\left[J_{r_e}(\tau_e) - J_{r_e}(\tau_K) - J_{r_{\phi^K}}(\tau_e) + J_{r_{\phi^K}}(\tau_K)\right] \\ &+ \mathbb{E}_{\tau_e, \tau_K}\left[J_{r_{\phi^K}}(\tau_e) - J_{r_{\phi^K}}(\tau_K)\right].\end{aligned}$$

The second term equals

$$J_{r_{\phi^K}}(\pi_e) - J_{r_{\phi^K}}(\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)}).$$

Since $\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)}$ is the optimal policy corresponding to r_{ϕ^K},

$$J_{r_{\phi^K}}(\pi_e) - J_{r_{\phi^K}}(\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)}) \leq 0.$$

Therefore,

$$\begin{aligned}\mathrm{SubOptR}(\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)}) &\leq \mathbb{E}_{\tau_e, \tau_K}\left[\left|(J_{r_e}(\tau_e) - J_{r_e}(\tau_K)) - (J_{r_{\phi^K}}(\tau_e) - J_{r_{\phi^K}}(\tau_K))\right|\right] \\ &\leq C_\eta \mathbb{E}_{\Gamma_{e,K}}\left[\|h_{r_{\phi^K}}(\cdot \mid \Gamma) - h_{r_e}(\cdot \mid \Gamma)\|_1\right].\end{aligned}$$

By Jensen's inequality,

$$\begin{aligned}\mathrm{SubOptR}(\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)}) &\leq C_\eta \left(\mathbb{E}_{\Gamma_{e,K}}\left[\|h_{r_{\phi^K}}(\cdot \mid \Gamma) - h_{r_e}(\cdot \mid \Gamma)\|_1^2\right]\right)^{1/2} \\ &\leq \mathcal{O}\left(\sqrt{\frac{1}{Kn} \log \frac{N(\mathcal{F}, \|\cdot\|_\infty, 1/(Kn))}{\delta}}\right).\end{aligned}$$

This proves the reward suboptimality bound.

Analogously, the cumulative cost difference between $\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)}$ and π_e according to the true cost function c_e is defined as $SubOptC(\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)}) \triangleq |J_{c_e}(\pi_e) - J_{c_e}(\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)})|$, the proof process for $SubOptC(\pi_{\phi^K, \psi^K, \lambda^K(\phi^K, \psi^K)})$ is the same

as that of $SubOptR(\pi_{\phi^*,\psi^*,\lambda^*(\phi^*,\psi^*)})$. We can get $SubOptC(\pi_{\phi^*,\psi^*,\lambda^*(\phi^*,\psi^*)}) \leq \mathcal{O}(\sqrt{\frac{1}{Kn}\ln(\frac{N(\mathcal{F},\|\cdot\|_\infty,\frac{1}{Kn})}{\delta}})$. As π_e is the expert policy, we can conclude that $J_{c_e}(\pi_e) = 0$. Then we can conclude that:

$$|J_{c_e}(\pi_e) - J_{c_e}(\pi_{\phi^K,\psi^K,\lambda^K(\phi^K,\psi^K)})| \leq \mathcal{O}(\sqrt{\frac{1}{Kn}\ln(\frac{N(\mathcal{F},\|\cdot\|_\infty,\frac{1}{Kn})}{\delta}}))$$

$$|0 - J_{c_e}(\pi_{\phi^K,\psi^K,\lambda^K(\phi^K,\psi^K)})| \leq \mathcal{O}(\sqrt{\frac{1}{Kn}\ln(\frac{N(\mathcal{F},\|\cdot\|_\infty,\frac{1}{Kn})}{\delta}}))$$

$$|J_{c_e}(\pi_{\phi^K,\psi^K,\lambda^K(\phi^K,\psi^K)})| \leq \mathcal{O}(\sqrt{\frac{1}{Kn}\ln(\frac{N(\mathcal{F},\|\cdot\|_\infty,\frac{1}{Kn})}{\delta}})).$$

□

L Experiment details and results

All Python3 codes are run on a Windows 10 desktop with 13th Gen Intel(R) Core(TM) i7-13700KF CPU and 32 GB of RAM. In CB-PbRL, the reward function updates every 10 environment set with three sets of feedback. We use SAC to calculate the policy, and the hyperparameters of SAC are shown below.

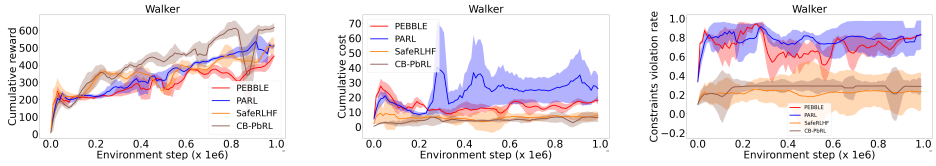
To improve the efficiency of our method in high-dimensional experiments, we will incorporate the following ideas. First, we will update the inner-level variable only once per outer iteration, thereby avoiding repeated inner-loop optimization and reducing the algorithm to a single-loop format. Second, we will employ first-order gradient estimators in place of second-order derivatives, substantially lowering the computational cost. These modifications can significantly improve scalability and make CB-PbRL more suitable for high-dimensional settings.

Hyperparameter	**Value**
Learning rate	0.001
Critic target update freq	10
(β_1, β_2)	(.9, .999)
Hidden units per each layer	256
Batch Size	100
Optimizer	Adam
Critic EMA τ	0.005
Discount γ	.99

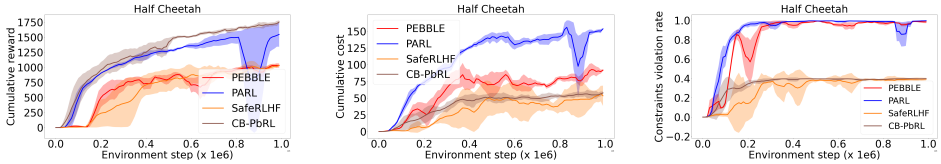
In the MuJoCo and Meta-World environments, the agent is incentivized to move as fast as possible or reach the goal quickly, which naturally encourages large actions. We therefore introduce a cost function that bounds the action magnitude. Consistent with this, PEBBLE exhibits a high constraint-violation rate, indicating that reward-maximizing behavior often exceeds this bound. This shows that maximizing the ground-truth reward alone does not automatically ensure constraint satisfaction. The detailed cost-function setup is provided in the following sections.

L.1 Wlaker2D

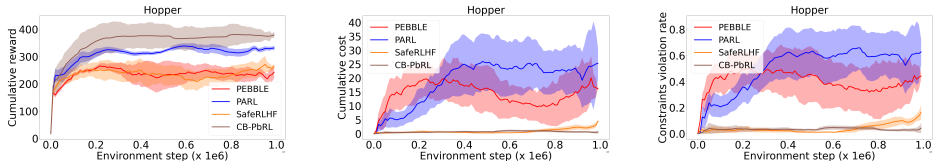
The walker is a two-dimensional bipedal robot consisting of a torso and two legs, each with a thigh, shin, and foot joint. Its movement is controlled by applying torques to these joints. The continuous observation and action spaces have dimensions of 17 and 6, respectively. The walker's objective is to move forward as quickly as possible while maintaining stability. The default reward function is the sum of a healthy reward, a forward reward, and a control cost. The healthy reward is a fixed value of 1 per timestep if the walker remains upright. The forward reward is proportional to the forward displacement, given by $125(x - x')$, where x and x' denote the walker's positions after and before the action, respectively. The control cost is $0.001\|a_w\|_2^2$, where a_w is the action. In addition, we impose a constraint on the angles of the torso and six joints, applying penalties when their absolute



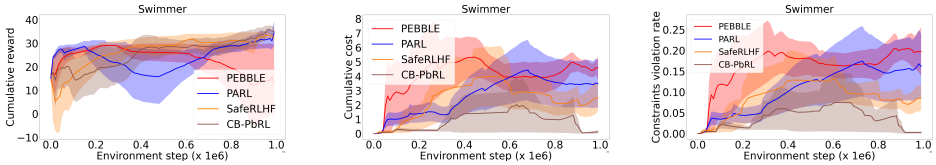
(a) Results on Walker2D. **Left**: Cumulative reward. **Middle** : Cumulative cost. **Right**: Constraint violation rate.



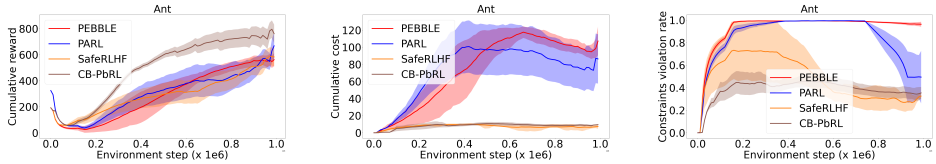
(b) Results on HalfCheetah. **Left**: Cumulative reward. **Middle** : Cumulative cost. **Right**: Constraint violation rate.



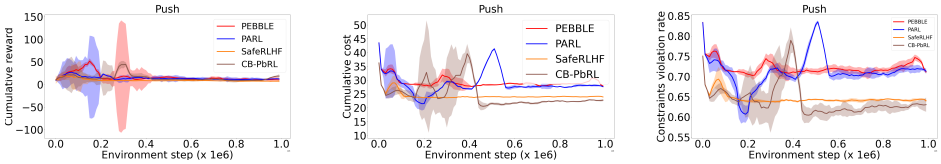
(c) Results on Hopper. **Left**: Cumulative reward. **Middle** : Cumulative cost. **Right**: Constraint violation rate.



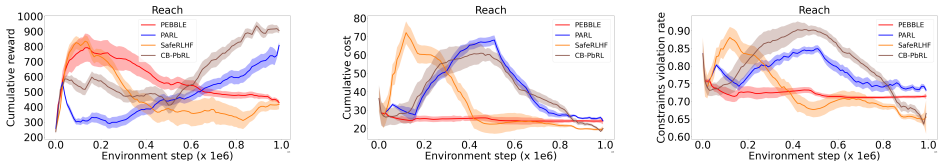
(d) Results on Swimmer. **Left**: Cumulative reward. **Middle** : Cumulative cost. **Right**: Constraint violation rate.



(e) Results on Ant. **Left**: Cumulative reward. **Middle** : Cumulative cost. **Right**: Constraint violation rate.



(f) Results on Push. **Left**: Cumulative reward. **Middle** : Cumulative cost. **Right**: Constraint violation rate.



(g) Results on Reach. **Left**: Cumulative reward. **Middle** : Cumulative cost. **Right**: Constraint violation rate.

values exceed 0.75 radians. The cost function is $\sum_{i=1}^{7} \max(0, |ag_i| - 0.75)$, where ag_i denotes the angle of the torso or a joint.

L.2 Half Cheetah

The Half Cheetah is a 2-dimensional robot with 8 joints. Its movement is controlled by applying torques to these joints. The continuous observation and action spaces have dimensions of 17 and 6, respectively. The robot's objective is to move forward as quickly as possible. The default reward function is the sum of a forward reward, and a control cost. The forward reward is proportional to the forward displacement, given by $20(x - x')$, where x and x' denote the walker's positions after and before the action, respectively. The control cost is $0.001\|a_w\|_2^2$, where a_w is the action. In addition, we impose a constraint on the angular velocity of 7 hinges, applying penalties when their absolute values exceed 12 radians per second. The cost function is $\sum_{i=1}^{7} \max(0, |av_i| - 12)$, where av_i denotes the angular velocity of a hinge.

L.3 Hooper

The hopper is a two-dimensional one-legged figure 4 body parts: torso, thigh, leg, and foot. Its movement is controlled by applying torques to three hinges. The continuous observation and action spaces have dimensions of 11 and 3, respectively. The robot's objective is to move forward as quickly as possible. The default reward function is the same as that of the Walker2D. In addition, we impose a constraint on the torques of hinges, applying penalties when their absolute values exceed 0.75 Nm. The cost function is $\sum_{i=1}^{3} \max(0, |t_i| - 0.75)$, where t_i denotes the torque of a hinge.

L.4 Swimmer

The swimmer is a planar 2D agent with a torso and two linked segments. Its movement is controlled by applying torques to the rotors and using fluid friction. The continuous observation and action spaces have dimensions of 8 and 2, respectively. The robot's objective is to move as fast as possible towards the right. The default reward function is the sum of a forward reward, and a control cost. The forward reward is proportional to the forward displacement, given by $25(x - x')$, where x and x' denote the walker's positions after and before the action, respectively. The control cost is $0.0001\|a_w\|_2^2$, where a_w is the action. In addition, we impose a constraint on the angular velocity of 2 hinges, applying penalties when their absolute values exceed 0.75 radians per second. The cost function is $\sum_{i=1}^{2} \max(0, |t_i| - 0.75)$, where t_i denotes the angular velocity of a hinge.

L.5 Ant

The Ant is a 3D quadruped robot composed of a torso, modeled as a freely rotating rigid body, with four legs attached. Each leg consists of two segments connected by joints. The robot's objective is to coordinate all four legs to move forward by applying torques at the eight hinge joints. In addition, we impose a constraint on the torque of 8 hinge joints, applying penalties when their absolute values exceed 0.75 Nm. The cost function is $\sum_{i=1}^{8} \max(0, |t_i| - 0.75)$, where t_i denotes the torque of a hinge.

L.6 Push and Reach

These two tasks are implemented using the Sawyer robotic arm in the Meta-World benchmark suite. In the Push task, the robot must push a cylindrical puck to a designated target location on the table surface. In the Reach task, the robot is required to move its end-effector to a specified goal position in 3D space. The robot's action space consists of four continuous control inputs: Cartesian displacements along the x, y, and z axes, and an additional scalar controlling the gripper. To encourage smooth and constrained movements, we impose an action penalty when the absolute value of any control input exceeds 0.75. The cost function is $\sum_{i=1}^{4} \max(0, |t_i| - 0.75)$.

M Limitation

Exact second-order hypergradient computation can be expensive for very large models such as LLMs. Our paper does not claim immediate LLM-scale applicability; instead, it focuses on formulating constrained bilevel PbRL and validating it in continuous-control settings. In large-scale regimes, bilevel methods can be implemented in a single-loop form with first-order hypergradient estimators, where $\lambda^*(\phi, \psi)$ is updated once per outer iteration and explicit second-order derivatives are avoided. We will consider improving scalability and making CB-PbRL more suitable for high-dimensional settings as our future work. According to the feedback setting, our method requires richer feedback than the standard one-dimensional preference learning.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction include the contribution and the scope of developing an algorithm for interactive inverse reinforcement learning.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

Justification: The limitation is listed in the Appendix Section M.

Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The proofs are shown in the Appendix.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: All information is shown in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

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

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

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

   Answer: [Yes]

   Justification: The paper provides open access to the data and code.

   Guidelines:

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

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Details are shown in Appendix Section L.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The explanation of the error bar is shown in the Appendix Section L.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Details are shown in Appendix Section L.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research in this paper follows the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: No societal impact of the work performed.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer:[Yes]

    Justification: The used assets are properly credited. The license and terms of use are explicitly mentioned and properly respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The LLM is used only for writing, editing, or formatting purposes.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.