UTILIZING EXPLAINABLE REINFORCEMENT LEARNING TO IMPROVE REINFORCEMENT LEARNING: A THEORETICAL AND SYSTEMATIC FRAMEWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) faces two challenges: (1) The RL agent lacks explainability. (2) The trained RL agent is, in many cases, non-optimal and even far from optimal. To address the first challenge, explainable reinforcement learning (XRL) is proposed to explain the decision-making of the RL agent. In this paper, we demonstrate that XRL can also be used to address the second challenge, i.e., improve RL performance. Our method has two parts. The first part provides a two-level explanation for why the RL agent is not optimal by identifying the mistakes made by the RL agent. Since this explanation includes the mistakes of the RL agent, it has the potential to help correct the mistakes and thus improve RL performance. The second part formulates a constrained bi-level optimization problem to learn how to best utilize the two-level explanation to improve RL performance. In specific, the upper level learns how to use the high-level explanation to shape the reward so that the corresponding policy can maximize the cumulative ground truth reward, and the lower level learns the corresponding policy by solving a constrained RL problem formulated using the low-level explanation. We propose a novel algorithm to solve this constrained bi-level optimization problem, and theoretically guarantee that the algorithm attains global optimality. We use MuJoCo experiments to show that our method outperforms state-of-the-art baselines.

1 INTRODUCTION

While reinforcement learning (RL) has been implemented in a wide range of applications, it faces two significant challenges: (1) The RL agent lacks transparency due to its black-box nature. (2) It has been widely observed in the RL community (Haarnoja et al., 2018; Henderson et al., 2018; Dulac-Arnold et al., 2019; Cheng et al., 2024) that, in many cases, the trained RL agent does not achieve maximum cumulative reward (i.e., non-optimal and even far from optimal). These two challenges motivate the need to improve the transparency and the performance of the RL agent.

To address the first challenge, explainable reinforcement learning (XRL) methods are proposed to explain the decision-making of the RL agents, including learning an interpretable policy (Bastani et al., 2018; Bewley & Lawry, 2021; Verma et al., 2018), pinpointing regions in the observations that are critical for choosing certain actions (Atrey et al., 2019; Guo et al., 2021a; Puri et al., 2019), learning the reward function that is actually maximized (Xie et al., 2022), and identifying the critical states that are influential to the cumulative reward (Guo et al., 2021b; Cheng et al., 2023; Amir & Amir, 2018). These XRL methods generate various explanations that improve the transparency of the RL agent and help people build trust in the RL agent (Vouros, 2022; Milani et al., 2023).

This paper demonstrates that XRL can also be used to address the second challenge of RL, i.e., RL improvement. Given a non-optimal RL agent, we use XRL to explain why this RL agent is not optimal by finding the mistakes made by the RL agent. Since our explanation provides insights into the RL agent's mistakes, it has the potential to help correct the mistakes and thus improve performance. Some recent works (Guo et al., 2021b; Cheng et al., 2023; 2024) also use XRL to improve the RL performance. In specific, they propose to first identify the critical states that are most influential to the cumulative reward as an explanation, and then perturb the actions (Guo et al., 2021b) or fine-tune the policy (Cheng et al., 2023; 2024) at those critical states such that the refined policy achieves higher cumulative reward. However, they do not explain why the RL agent does not maximize the cumulative reward. This paper proposes a novel framework that first explains why the RL agent is not optimal, and then learns how to utilize the generated explanations to improve RL performance. We summarize our contributions as follows:

Contribution statement. This paper proposes an optimization-based framework that aims to learn how to best utilize XRL to improve RL. We refer to this framework as "**ut**ilizing expla**i**nable R**L** to **i**mprove reinforcemen**t** learning efficac**y**" (UTILITY). Our contributions are threefold:

First, we provide a two-level explanation for why the RL agent is not optimal. The high-level explanation learns a reward function to which the RL agent is actually optimal, and then explains why the RL agent is not optimal by comparing this learned reward function to the ground truth reward function. The low-level explanation identifies the state-action pairs that lead the RL agent to be non-optimal. We refer to these state-action pairs as "misleading" state-action pairs, and rigorously derive a mathematical metric to identify the "misleading" state-action pairs as the low-level explanation.

Second, we mathematically formalize the problem of utilizing the two-level explanations to improve the RL performance as a constrained bi-level optimization problem. In specific, the upper-level problem aims to learn how to use the high-level explanation (i.e., the learned reward function) to shape the ground truth reward function to help the corresponding policy maximize the cumulative (ground truth) reward. The lower-level problem learns the corresponding policy by solving a constrained RL problem where the objective is to maximize the cumulative shaping reward and the constraint is to discourage from visiting the low-level explanation (i.e., the "misleading" state-action pairs). Current state-of-the-arts (Xu & Zhu, 2023; Khanduri et al., 2023) on constrained bi-level optimization can only deal with the case where the lower-level problem is strongly convex. However, in our case, both the objective function and constraint in the lower-level problem are highly non-convex. Therefore, a novel theoretical framework is desired to solve this constrained bi-level optimization problem.

Third, we develop a novel theoretical framework and thereby an algorithm to solve the constrained bi-level optimization problem. In specific, we first use a dual method to transform the constrained bi-level optimization problem to an equivalent unconstrained bi-level optimization problem, and then propose an approximation-based triple-loop algorithm to solve this unconstrained bi-level optimization problem. We quantify the approximation error at each loop and prove that the algorithm attains global optimality. Experiments show that UTILITY outperforms state-of-the-art baselines.

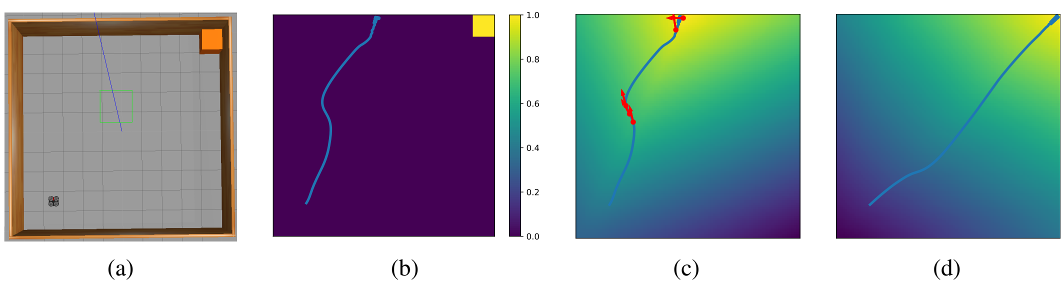


Figure 1: (a) An RL task where a drone starts from the lower-left corner and navigates to the orange goal at the upper-right corner. (b) A failing trajectory (blue) generated by the RL agent and a heat map that visualizes the ground truth reward. (c) The two-level explanation of why the blue trajectory fails to reach the goal. The high-level explanation is the learned reward (visualized as the heat map) to which the RL agent is actually optimal. The low-level explanation is the misleading state-action pairs (red circles and linked red arrows). (d) The trajectory after improvement reaches the goal.

Illustrative example. Figure 1 uses an example to illustrate our proposed framework. Suppose we use RL to navigate a drone to the orange goal in Figure 1a. The state is the 2-D coordinate and the action is the moving direction. The ground truth reward is one at the goal states and zero otherwise. Figure 1b uses a heat map to visualize the ground truth reward and the blue trajectory is generated by the learned policy. This learned policy is not optimal because it fails to reach the goal.

Figure 1c visualizes our two-level explanation of why the learned policy is not optimal. At the high level, we use the heat map to visualize the learned reward function to which the RL agent's trajectory (policy) is actually optimal. The high-level explanation is that the RL agent's policy is actually optimal to the learned reward function (visualized as the heat map in Figure 1c), and this learned reward function is very different from the ground truth reward function (visualized as the heat map in Figure 1b). Note that we normalize all the reward functions in Figures 1b-1d to $[0, 1]$ for better comparison. For the low-level explanation, we identify the top five "misleading" state-action pairs (i.e., the red circles and arrows) in the blue trajectory where the red circles are the states and the linked red arrows are the corresponding actions chosen by the non-optimal RL agent. These state-action pairs are "misleading" since the correct actions should point to the goal.

Figure 1d shows the improvement where the heat map visualizes the learned shaping reward function and the blue trajectory is generated by the learned policy after improvement. We can see that the learned policy after improvement successfully reaches the goal.

Note that Figure 1 is just for illustration to help understand the framework. We are aware that current RL algorithms can succeed the task in Figure 1a without improvement, and we use more complicated tasks in the experiment (Section 5) to show the improvement of the proposed framework.

2 Related works

Due to the space limit, we only include the related works on improving RL performance here, and we include more related works in Appendix E.

Reward shaping. Reward shaping can improve the RL performance by shaping the ground truth reward function. Current works on reward shaping has two main categories. The first category (Ng et al., 1999; Hu et al., 2020; Devlin & Kudenko, 2012; Gupta et al., 2022) requires an external source, such as a human expert, to provide domain knowledge as an ingredient to shape the ground truth reward function. However, when the tasks become complicated, it could be difficult and even infeasible for humans to provide domain knowledge. The second category does not need domain knowledge, including reward shaping based on exploration bonus (Bellemare et al., 2016; Ostrovski et al., 2017), learning an intrinsic reward (Zheng et al., 2018; Memarian et al., 2021), and combining exploration bonus and intrinsic reward (Devidze et al., 2022). The first category usually has better performance, while the second category does not require human-domain knowledge.

Other methods that can improve RL performance. Lazy-MDP (Jacq et al., 2022) shows performance improvement with the help of a provided default policy. It uses the "lazy-gap" to determine whether to choose greedy action or follow a default policy on each state s. Self-imitation learning (Oh et al., 2018) aims to encourage deep exploration by reproducing previous good decisions. Papers (Wang & Taylor, 2017; Taylor, 2018; Taylor et al., 2023) aim to improve the RL performance by utilizing external assistance, such as the assistance of a pre-trained RL agent (Wang & Taylor, 2017) or a human (Taylor, 2018; Taylor et al., 2023), which may not be accessible in some scenarios.

3 Two-level explanation of why the RL agent is non-optimal

This section provides a two-level explanation to explain why the RL agent is not optimal. The RL agent's decision making is based on a Markov decision process (MDP) $(\mathcal{S}, \mathcal{A}, \gamma, P_0, P, r)$ which consists of a state set $\mathcal{S}$, an action set $\mathcal{A}$, a discount factor $\gamma \in (0, 1)$, an initial state distribution $P_0(\cdot)$, a state transition function $P(\cdot|\cdot,\cdot)$, and the ground truth reward function $r(\cdot,\cdot)$. The RL agent's learned policy is denoted by π_A and the cumulative reward is defined as $J_r(\pi) \triangleq E^{\pi}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$ where the initial state is drawn from P_0. When we say that the RL agent is not optimal, it means that $\pi_A \notin \arg\max_{\pi} J_r(\pi)$.

The black-box assumption. To ensure practicability, following (Bewley & Lawry, 2021; Guo et al., 2021b; Cheng et al., 2023; Guidotti et al., 2019), we only treat the RL agent as a black box with no access to its internal structure. In specific, we do not assume the access to the learned value/Q-function nor the learned policy π_A of the RL agent. We can only observe a set of m trajectories $\mathcal{D} \triangleq \{\zeta^j\}_{j=1}^m$ demonstrated by the RL agent (using the non-optimal policy π_A) where each trajectory $\zeta^j = s_0, a_0, \cdots$ is a state-action sequence.

The high-level explanation. At a high level, since the RL agent is not optimal to the ground truth reward function r, we can learn a reward function $\hat{r}$ to which the RL agent's policy π_A is actually optimal, and use this learned reward function $\hat{r}$ to generate explanations. A recent work (Xie et al., 2022) uses the state-action pairs (s, a) with the highest $\hat{r}(s, a)$ as an explanation, however, these state-action pairs cannot explain why the RL agent is not optimal. Therefore, we extend (Xie et al., 2022) by comparing the learned reward function $\hat{r}$ to the ground truth reward function r to explain why π_A is not optimal to the ground truth r. Figures 1b-1c provide an example of our high-level explanation: the policy π_A is actually optimal to the learned reward function $\hat{r}$ (visualized in Figure 1c), and this learned reward function $\hat{r}$ is very different from the ground truth reward function r (visualized in Figure 1b).

Inverse reinforcement learning (IRL) (Abbeel & Ng, 2004; Ziebart et al., 2008; Arora & Doshi, 2021) can learn the reward function $\hat{r}$ and an associated policy $\hat{\pi}_A$ from the demonstration set $\mathcal{D}$ such that the behaviors of policy π_A demonstrated in $\mathcal{D}$ are optimal to the reward function $\hat{r}$ learned by IRL, and the learned policy $\hat{\pi}_A$ can imitate the policy π_A. We use maximum likelihood IRL (Zeng et al., 2022) to learn the reward function $\hat{r}$ and policy $\hat{\pi}_A$.

While the learned reward function $\hat{r}$ can be used for the high-level explanation, it is only interpretable to humans in low dimension, e.g., we can use heat maps to plot reward functions (as in Figure 1). When the state and action become high dimensional, the learned reward function $\hat{r}$ is hard for humans to understand and thus it is difficult to straightforwardly compare $\hat{r}$ to r (as we did in Figures 1b-1c). Therefore, we need the low-level explanation which is still interpretable in high dimension.

The low-level explanation. At a low level, the RL agent is not optimal meaning that it visits some critical points that lead to the non-optimality. Recent works (Guo et al., 2021b; Cheng et al., 2023; Amir & Amir, 2018; Jacq et al., 2022) identify the states that are most influential to the cumulative reward as critical points. In order to explain why the RL agent is not optimal, we extend their idea by redefining the critical points as the state-action pairs that lead π_A to be non-optimal. We refer to these critical points as "misleading" state-action pairs and we aim to identify the top K "misleading" state-action pairs in the demonstration set $\mathcal{D}$ as the low-level explanation. Note that we use infinite time-horizon MDP and it is not possible to identify the top K "misleading" state-pairs if a trajectory has infinitely many different state-action pairs. However, in practice, the trajectory length is usually finite and we can unify the notions of finite time horizon and infinite time horizon by introducing "absorbing state" (Sutton & Barto, 2018). In specific, we can treat the terminal state of a finite-time-horizon trajectory as a state keeping transitioning only to itself and generating zero reward.

The *key challenge* to identify the top K "misleading" state-action pairs in the demonstration set $\mathcal{D}$ is to propose a proper criterion or metric to define what a "misleading" state-action pair is. A straightforward way is to identify the state-action pairs that an optimal policy will not visit as "misleading", and thus use $\pi_A(a|s) - \pi^*(a|s)$ as the metric where π^* is an optimal policy. However, this metric is infeasible because the optimal policy π^* is not accessible.

In contrast, we derive a feasible metric in Definition 1 that uses a Q-function to find misleading state-action pairs in the demonstration set $\mathcal{D}$. The Q-function under the policy π and reward function r is $Q_r^\pi(s, a) \triangleq E^\pi[\sum_{t=0}^\infty \gamma^t r(s_t, a_t)|s_0 = s, a_0 = a]$.

Definition 1. *A state-action pair $(s, a) \in \mathcal{D}$ is a misleading state-action pair if $l(s, a) > 0$ where $l(s, a) \triangleq \max_{a'} Q_r^{\pi_A}(s, a') - Q_r^{\pi_A}(s, a)$ is referred to as "misleading level". The larger the misleading level $l(s, a)$ is, the more misleading the state-action pair (s, a) is.*

We include the derivation of how we come up with this metric l and the proof of why $(s, a) \in \mathcal{D}$ is misleading if $l(s, a) > 0$ in Appendix B.1. In brief, we prove in Appendix B.1 that the policy π_A will be an optimal policy if $l(s, a) = 0$ for all (s, a) such that $a \in \pi_A(s)$ where $\pi_A(s)$ is the set of actions that the policy π_A has nonzero probability to choose at the state s. Therefore, any state-action pair $(s, a) \in \mathcal{D}$ such that $l(s, a) > 0$ can be regarded as a "misleading" state-action pair that leads the policy π_A to be non-optimal. The larger the misleading level $l(s, a)$ is, the more "misleading" the state-action pair (s, a) is, because the Q value of the chosen action a has a larger gap from the maximum Q value at the state s. We denote the set of the identified top K "misleading" state-action pairs by $\mathcal{C}$, which serves as the low-level explanation.

While we cannot access the policy π_A, we have already learned the policy $\hat{\pi}_A$ using IRL and the policy $\hat{\pi}_A$ imitates the policy π_A. Therefore, we can use $Q_r^{\hat{\pi}_A}$ to substitute for $Q_r^{\pi_A}$. Given that we can access $\hat{\pi}_A$ and r, we can simply learn $Q_r^{\hat{\pi}_A}$ by sampling the environment to collect enough data and doing regression. Due to the space limit, we include the method of learning $Q_r^{\hat{\pi}_A}$ in Appendix

B.2. We are aware that in RL, it is usually sample inefficient and computationally expensive if we want to sample enough data to learn precise Q-functions corresponding to all the learning policy in the learning procedure. However, our case is different because we only need to learn one precise Q-function, which corresponds to the specific policy $\hat{\pi}_A$.

4 UTILIZING THE TWO-LEVEL EXPLANATION TO IMPROVE RL

This section provides a theoretical framework that utilizes the two-level explanation in Section 3 to improve the RL performance. In specific, Subsection 4.1 formulates the problem as a constrained bi-level optimization problem. Subsection 4.2 proposes a novel theoretical framework and thereby an algorithm to solve the constrained bi-level optimization problem.

4.1 PROBLEM FORMULATION

We aim to utilize the two-level explanation to improve the RL performance. For the high-level explanation $\hat{r}$, we use it to formulate a domain knowledge and learn how to use this domain knowledge to shape the ground truth reward r such that the learned shaping reward can lead the policy to maximize the cumulative ground truth reward $J_r(\pi)$. In specific, we use the comparison $r - \hat{r}$ between the ground truth reward r and the high-level explanation $\hat{r}$ as the domain knowledge. Note that in practice, we need to first scale $\hat{r}$ to the same scale with r for a better comparison. Since this comparison quantifies the RL agent's misunderstanding of the ground truth reward function r, it has the potential to help patch the error. Towards this end, we propose to learn a shaping reward function $r_\theta(\cdot, \cdot)$ parameterized by θ, which takes the original reward and the domain knowledge as the input. For a given state-action pair (s, a), the corresponding shaping reward is $r_\theta(r(s,a), r(s,a) - \hat{r}(s,a))$.

For the low-level explanation $\mathcal{C}$, we discourage the RL agent from visiting the "misleading" state-action pairs in $\mathcal{C}$. Towards this end, we design a cost function $c(\cdot, \cdot)$ such that $c(s,a) \in (0, c_{\max}]$ when $(s,a) \in \mathcal{C}$, and $c(s,a) = 0$ otherwise, where $c_{\max}$ is a positive constant. We discourage from visiting $\mathcal{C}$ by constraining the cumulative cost $J_c(\pi) \triangleq E^\pi[\sum_{t=0}^\infty \gamma^t c(s_t, a_t)]$ under a budget b.

Remark on why discouraging misleading state-action pairs can improve π_A. According to monotonic policy improvement theorem, π_A will improve if it chooses greedy actions according to its Q-function $Q_r^{\pi_A}$ at all the states. Given that π_A stops improving, it means that π_A must choose some nongreedy actions at some states, i.e., the misleading state-action pairs. Constraining these misleading state-action pairs means that we constrain the nongreedy actions π_A originally chooses at the states. This constraint can help π_A choose greedy actions because it eliminates some nongreedy actions and thus π_A only needs to find greedy actions from smaller action sets. Since this constraint can help π_A find greedy actions, it can help improve π_A.

To utilize the two-level explanation, we formulate a constrained bi-level optimization problem:

$$\max_\theta \; J_r(\pi_{r_\theta}), \text{ where } \pi_{r_\theta} = \arg\max_\pi \{J_{r_\theta}(\pi) + H(\pi), \text{ s.t. } J_c(\pi) \leq b\}, \tag{1}$$

where $J_{r_\theta}(\pi) \triangleq E^\pi[\sum_{t=0}^\infty \gamma^t r_\theta(r(s_t,a_t), r(s_t,a_t) - \hat{r}(s_t,a_t))]$ is the cumulative shaping reward, and the causal entropy $H(\pi) \triangleq E^\pi[\sum_{t=0}^\infty -\gamma^t \log \pi(a_t|s_t)]$ is to encourage exploration and is widely used in soft Q-learning (Haarnoja et al., 2017) and soft actor-critic (Haarnoja et al., 2018).

In the problem (1), the upper level aims to learn a shaping reward function r_θ such that the corresponding policy π_{r_θ} can achieve maximum cumulative ground truth reward $J_r(\pi_{r_\theta})$. Given the current learned shaping reward function r_θ, the lower-level problem in (1) is to compute the corresponding policy π_{r_θ} by solving a constrained RL problem. The constrained RL problem encourages π_{r_θ} to maximize the entropy-regularized cumulative shaping reward $(J_{r_\theta}(\pi) + H(\pi))$ and discourages π_{r_θ} from visiting $\mathcal{C}$ by controlling the cumulative cost $J_c(\pi)$ under the budget b. Note that if we choose $b = 0$, it means that the policy π_{r_θ} should totally avoid the set $\mathcal{C}$.

Before solving the problem (1), we need to first make sure that the problem (1) is well-defined. In specific, since the lower-level problem in (1) is non-convex, it may have more than one optimal solution, i.e., π_{r_θ} is not unique. Therefore, given a reward parameter θ, the corresponding upper-level objective function value $J_r(\pi_{r_\theta})$ may not be unique as π_{r_θ} is not unique. This will make the problem (1) ill-defined. The following theorem guarantees that the problem (1) is well-defined.

Theorem 1. *Given reward r_θ, the optimal solution π_{r_θ} of the lower-level problem in (1) is unique.*

4.2 THEORETICAL FRAMEWORK

While the current state-of-the-arts (Xu & Zhu, 2023; Khanduri et al., 2023) on constrained bi-level optimization can only deal with strongly convex lower-level problems, both the objective function and the constraint of the lower-level problem in (1) are non-convex. Therefore, a novel theoretical framework is desired to solve the problem (1). This subsection proposes a novel theoretical framework to solve the problem (1).

The proposed theoretical framework has three parts. (i) The first part transforms the original constrained bi-level optimization problem (1) to an equivalent unconstrained bi-level optimization problem. The benefit of this transformation is that the equivalent unconstrained bi-level optimization problem has an unconstrained and convex lower-level problem, which is more tractable and easier to solve. (ii) The second part proposes a novel algorithm to solve the problem (1) by solving the equivalent unconstrained bi-level optimization problem. (iii) The third part theoretically guarantees that the proposed algorithm attains global optimality.

4.2.1 PROBLEM TRANSFORMATION

The lower-level problem of the problem (1) is non-convex. To deal with the non-convexity issue, we introduce the dual function of the lower-level problem in (1): $G(\lambda;\theta) \triangleq \max_\pi J_{r_\theta}(\pi) + H(\pi) - \lambda(J_c(\pi) - b)$ where λ is the dual variable. The dual function $G(\lambda;\theta)$ is convex in λ since it is the point-wise maximum over a set of affine functions of λ (Boyd & Vandenberghe, 2004).

Theorem 2. *The optimal solution of the lower-level problem in (1) is uniquely the constrained soft policy $\pi_{\lambda^*(\theta);\theta}$ where $\lambda^*(\theta)$ is the unique optimal solution of the dual problem $\min_\lambda G(\lambda;\theta)$.*

We include the analytical expression of the constrained soft policy $\pi_{\lambda^*(\theta);\theta}$ (Liu & Zhu, 2022) in Appendix C. Theorem 2 indicates that $\pi_{\lambda^*(\theta);\theta}$ is the unique optimal solution of the lower-level problem in (1) (i.e., $\pi_{\lambda^*(\theta);\theta} = \pi_{r_\theta}$), and $\lambda^*(\theta) = \arg\min_\lambda G(\lambda;\theta)$. Therefore, we can replace π_{r_θ} with $\pi_{\lambda^*(\theta);\theta}$ and replace the lower-level problem in (1) with its dual problem, and thereby transform the constrained bi-level optimization problem (1) to the following unconstrained bi-level optimization problem:

$$\max_\theta \ J_r(\pi_{\lambda^*(\theta);\theta}), \text{ where } \lambda^*(\theta) = \arg\min_\lambda G(\lambda;\theta). \tag{2}$$

Compared to the original problem (1), the lower-level problem of the problem (2) is unconstrained and convex. However, there are still two challenges to solve the new problem (2).

Challenge (i): Evaluating the dual function $G(\lambda;\theta)$ needs to obtain the constrained soft policy $\pi_{\lambda;\theta} = \arg\max_\pi J_{r_\theta}(\pi) + H(\pi) - \lambda(J_c(\pi) - b)$. However, current RL algorithms can only approach $\pi_{\lambda;\theta}$ at a certain rate and only obtain the exact $\pi_{\lambda;\theta}$ when iteration number goes to infinity. In practice, we can only run an algorithm for finite iterations and thus we cannot obtain the exact $\pi_{\lambda;\theta}$. This will cause errors when we evaluate the dual function G.

Challenge (ii): Even if we can obtain the exact $\pi_{\lambda;\theta}$, we cannot guarantee to get the exact optimal solution $\lambda^*(\theta)$ of the lower-level problem in finite time. This makes it difficult to evaluate and solve the upper-level problem in (2) since the upper-level problem in (2) requires $\lambda^*(\theta)$.

4.2.2 THE PROPOSED ALGORITHM

This part proposes a novel algorithm that solves problem (1) by solving problem (2). The proposed algorithm is triple-loop where the inner loop approximates the constrained soft policy $\pi_{\lambda;\theta}$ and tackles Challenge (i), the middle loop approximates the optimal solution $\lambda^*(\theta)$ of the lower-level problem in (2) and tackles Challenge (ii), and the outer loop solves the upper-level problem in (2). We use n, $\bar{n}$, and $\tilde{n}$ to respectively denote the iteration indices of outer, middle, and inner loop.

Algorithm 1 first generates the two-level explanation (line 1) and then uses three loops to utilize the generated two-level explanation. In specific, the inner loop (lines 4-7) approximates the constrained soft policy $\pi_{\lambda;\theta}$. With the approximated policy $\hat{\pi}_{\lambda;\theta}$ (line 8), the middle loop solves the lower-level problem in (2) via ($\bar{N} - 1$)-step gradient descent (line 9) to approximate the optimal solution $\lambda^*(\theta)$. With the approximated parameter $\hat{\lambda}(\theta)$ (line 11), the outer loop solves the upper-level problem in (2) via ($N - 1$)-step gradient ascent (line 12). In the following, we elaborate each loop respectively.

Algorithm 1 Utilizing explainable reinforcement learning to improve reinforcement learning

Input: Demonstration set $\mathcal{D}$, initial shaping reward parameter θ_0, dual parameter λ_0, and policy π_0
Output: Shaping reward r_{θ_N} and the policy after improvement $\hat{\pi}_{\hat{\lambda}(\theta_N);\theta_N}$

Generate the two-level explanation $(\hat{r}, \mathcal{C})$
for $n = 0, \cdots, N-1$ **do**
  for $\bar{n} = 0, \cdots, \bar{N}-1$ **do**
    for $\tilde{n} = 0, \cdots, \tilde{N}_{\bar{n}} - 1$ **do**
      Compute the constrained soft Q function $Q^{\pi_{\tilde{n}}}_{\lambda_{\bar{n}};\theta_n}$
      Update the policy $\pi_{\tilde{n}+1}(a|s) \propto \exp(Q^{\pi_{\tilde{n}}}_{\lambda_{\bar{n}};\theta_n}(s,a))$ for any $(s,a) \in \mathcal{S} \times \mathcal{A}$
    end for
    Set $\hat{\pi}_{\lambda_{\bar{n}};\theta_n} = \pi_{\tilde{N}_{\bar{n}}}$ and use $\hat{\pi}_{\lambda_{\bar{n}};\theta_n}$ to compute the approximated gradient $g_{\lambda_n;\theta_n}$
    Update $\lambda_{\bar{n}+1} = \lambda_{\bar{n}} - \alpha_{\bar{n}} g_{\lambda_{\bar{n}};\theta_n}$
  end for
  Set $\hat{\lambda}(\theta_n) = \frac{1}{\bar{N}} \sum_{\bar{n}=0}^{\bar{N}-1} \lambda_{\bar{n}}$ and compute $\hat{\pi}_{\hat{\lambda}(\theta_n);\theta_n}$ via $(\bar{N}-1)$-step soft policy iteration
  Use $\hat{\pi}_{\hat{\lambda}(\theta_n);\theta_n}$ to compute the approximated gradient g_{θ_n} and update $\theta_{n+1} = \theta_n + \beta_n g_{\theta_n}$
end for

The inner loop. Given the parameter (λ, θ), the inner loop aims to approximate the constrained soft policy $\pi_{\lambda;\theta}$ via $\tilde{N}_{\bar{n}}$-step soft policy iteration (Haarnoja et al., 2017), and $\tilde{N}_{\bar{n}} = \bar{n} + 1$. Soft policy iteration has two steps: policy evaluation and policy improvement. Policy evaluation computes the constrained soft Q-function $Q^{\pi_{\tilde{n}}}_{\lambda;\theta}$ corresponding to the current policy $\pi_{\tilde{n}}$, dual parameter λ, and reward parameter θ. We include the expression of the constrained soft Q-function in Appendix C. Policy improvement aims to update the policy according to $\pi_{\tilde{n}+1}(a|s) \propto \exp(Q^{\pi_{\tilde{n}}}_{\lambda;\theta}(s,a))$ for any $(s,a) \in \mathcal{S} \times \mathcal{A}$. The output of the inner loop is the approximated policy $\hat{\pi}_{\lambda;\theta} = \pi_{\tilde{N}_{\bar{n}}}$. In practical implementations, we can update the policy $\pi_{\tilde{n}}$ via the policy update in soft Q-learning (Haarnoja et al., 2017) or actor update in soft actor-critic (Haarnoja et al., 2018). While soft Q-learning and soft actor-critic are designed for unconstrained RL, we show in Appendix C that we can revise them to approximate the constrained soft policy.

The middle loop. We aim to solve the lower-level problem in (2) via ($\bar{N}$-1)-step gradient descent.

Lemma 1. *The gradient of the dual function G is $\nabla_\lambda G(\lambda;\theta) = b - J_c(\pi_{\lambda;\theta})$.*

The gradient $\nabla_\lambda G(\lambda;\theta)$ requires the exact constrained soft policy $\pi_{\lambda;\theta}$ which is inaccessible. Therefore, we use the approximated policy $\hat{\pi}_{\lambda;\theta}$ obtained from the inner loop to approximate the gradient $\nabla_\lambda G(\lambda;\theta)$ via the gradient approximation $g_{\lambda;\theta} = b - J_c(\hat{\pi}_{\lambda;\theta})$, and solve the lower-level problem via ($\bar{N}$-1)-step gradient descent $\lambda_{\bar{n}+1} = \lambda_{\bar{n}} - \alpha_{\bar{n}} g_{\lambda_{\bar{n}};\theta}$. The output is $\hat{\lambda}(\theta) = \frac{1}{\bar{N}} \sum_{\bar{n}=0}^{\bar{N}-1} \lambda_{\bar{n}}$.

The outer loop. We aim to solve the upper-level problem in (2) via $(N-1)$-step gradient ascent. Towards this end, we generalize the Q/value function (Lin et al., 2020; Sutton & Barto, 2018). In specific, we define the Q-function of cost c under policy π as $Q^\pi_c(s,a) \triangleq E^\pi[\sum_{t=0}^\infty \gamma^t c(s_t,a_t)|s_0 = s, a_0 = a]$ and value function of cost as $V^\pi_c(s) \triangleq E^\pi[\sum_{t=0}^\infty \gamma^t c(s_t,a_t)|s_0 = s]$. We define the Q-function of reward gradient $\nabla_\theta r_\theta$ as $Q^\pi_{\nabla_\theta r_\theta}(s,a) \triangleq E^\pi[\sum_{t=0}^\infty \gamma^t \nabla_\theta r_\theta(s_t,a_t)|s_0 = s, a_0 = a]$ and value function of reward gradient as $V^\pi_{\nabla_\theta r_\theta}(s) \triangleq E^\pi[\sum_{t=0}^\infty \gamma^t \nabla_\theta r_\theta(s_t,a_t)|s_0 = s]$. We define state-action visitation frequency as $\psi^\pi(s,a) \triangleq E^\pi[\sum_{t=0}^\pi \gamma^t \mathbb{1}\{s_t = s, a_t = a\}]$ where $\mathbb{1}\{\cdot\}$ is the indicator function.

Lemma 2. *The upper-level gradient is $dJ_r(\pi_{\lambda^*(\theta);\theta})/d\theta = E_{(s,a)\sim\psi^{\pi_{\lambda^*(\theta);\theta}}}\big[\big(Q^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s,a) - V^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s) - C_{\pi_{\lambda^*(\theta);\theta}}(Q^{\pi_{\lambda^*(\theta);\theta}}_c(s,a) - V^{\pi_{\lambda^*(\theta);\theta}}_c(s))\big)Q^{\pi_{\lambda^*(\theta);\theta}}_r(s,a)\big]$ where C_π is a constant vector if we fix policy π, and we include the expression of C_π in Appendix D.3.*

Since the gradient $\frac{dJ_r(\pi_{\lambda^*(\theta);\theta})}{d\theta}$ requires the exact optimal solution $\lambda^*(\theta)$ and the exact constrained soft policy $\pi_{\lambda^*(\theta);\theta}$, we can only use the policy $\hat{\pi}_{\hat{\lambda}(\theta);\theta}$ to approximate $\frac{dJ_r(\pi_{\lambda^*(\theta);\theta})}{d\theta}$ via $g_\theta = E_{(s,a)\sim\psi^{\hat{\pi}_{\hat{\lambda}(\theta);\theta}}}\big[\big(Q^{\hat{\pi}_{\hat{\lambda}(\theta);\theta}}_{\nabla_\theta r_\theta}(s,a) - V^{\hat{\pi}_{\hat{\lambda}(\theta);\theta}}_{\nabla_\theta r_\theta}(s) - C_{\hat{\pi}_{\hat{\lambda}(\theta);\theta}}(Q^{\hat{\pi}_{\hat{\lambda}(\theta);\theta}}_c(s,a) - V^{\hat{\pi}_{\hat{\lambda}(\theta);\theta}}_c(s))\big)Q^{\hat{\pi}_{\hat{\lambda}(\theta);\theta}}_r(s,a)\big]$. We then solve the upper-level problem in (2) via $(N-1)$-step gradient ascent $\theta_{n+1} = \theta_n + \beta_n g_{\theta_n}$.

4.2.3 THEORETICAL ANALYSIS

This part quantifies the optimality of the policy after improvement $\hat{\pi}_{\hat{\lambda}(\theta_N);\theta_N}$. The main difficulty is that the inner loop and middle loop can only approximate the policy $\pi_{\lambda;\theta}$ and the optimal solution $\lambda^*(\theta)$, and the approximation error may accumulate and ruin the convergence of the outer loop. In the following context, we sequentially quantify the convergence from the inner loop to the outer loop.

Lemma 3 (convergence of the inner loop)**.** *Given the parameter (λ, θ), the output $\hat{\pi}_{\lambda;\theta}$ of the inner loop satisfies $|\log \hat{\pi}_{\lambda;\theta}(a|s) - \log \pi_{\lambda;\theta}(a|s)| \leq O(\gamma^{\tilde{N}_{\bar{n}}})$ for any $(s, a) \in \mathcal{S} \times \mathcal{A}$.*

Lemma 3 shows that inner loop converges linearly to the exact constrained soft policy $\pi_{\lambda;\theta}$.

Assumption 1. *(i) It holds that $|r_\theta(\cdot, \cdot)| \leq C_1$ for any θ where C_1 is a positive constant. (ii) It holds that $||\nabla_\theta r_\theta(\cdot, \cdot)|| \leq C_2$ and $||\nabla^2_{\theta\theta} r_\theta(\cdot, \cdot)|| \leq C_3$, where C_2 and C_3 are some positive constants.*

Assumption 1 assumes that r_θ is bounded, Lipschitz continuous, and smooth to θ, which is a common assumption in RL (Wang et al., 2019; Kumar et al., 2023; Zhang et al., 2020). We next quantify the convergence of the middle loop:

Lemma 4 (convergence of the middle loop)**.** *Suppose Assumption 1 (ii) holds and let $\alpha_{\bar{n}} = 1/(\bar{n} + 1)^{\bar{\eta}}$ where $\bar{\eta} \in (1/2, 1)$, the outputs $(\hat{\lambda}(\theta), \hat{\pi}_{\hat{\lambda}(\theta);\theta})$ of the middle loop satisfy that (i) $|\hat{\lambda}(\theta) - \lambda^*(\theta)| \leq O(1/\bar{N}^{1-\bar{\eta}})$; (ii) $|\log \hat{\pi}_{\hat{\lambda}(\theta);\theta}(a|s) - \log \pi_{\lambda^*(\theta);\theta}(a|s)| \leq O(1/\bar{N}^{1-\bar{\eta}} + \gamma^{\bar{N}})$ for any $(s, a) \in \mathcal{S} \times \mathcal{A}$.*

Lemma 4 shows that if the iteration $\bar{N}$ of middle loop is sufficiently large, the approximation error of $\lambda^*(\theta)$ and $\pi_{\lambda^*(\theta);\theta}$ can be arbitrarily small. We next quantify the convergence of the outer loop:

Theorem 3 (convergence of the outer loop)**.** *Suppose Assumption 1 and the conditions in Lemma 4 hold and let $\beta_n = 1/(n + 1)^\eta$ where $\eta \in (1/2, 1)$, then it holds $\frac{1}{N}\sum_{n=0}^{N-1} ||\nabla J_r(\pi_{\lambda^*(\theta_n);\theta_n})||^2 \leq O(1/N^{1-\eta} + 1/\bar{N}^{2-2\bar{\eta}} + \gamma^{2\bar{N}})$.*

Theorem 3 shows that Algorithm 1 converges to stationarity when the iteration numbers N and $\bar{N}$ go to infinity. When the state-action space is finite and r_θ is linear, we have the following stronger result:

Theorem 4 (optimality of the outer loop)**.** *Suppose the conditions in Lemma 4 hold, the state-action space is finite, and r_θ is linear. Let the step size $\beta_n \leq \min\{(1-\gamma)^3/8, 1/\bar{L}\}$, then it holds that $\lim_{N\to\infty} \lim_{\bar{N}\to\infty} J_r(\hat{\pi}_{\hat{\lambda}(\theta_N);\theta_N}) - J^*_r = 0$ where J^*_r is the maximum value of $J_r(\pi)$, and $\bar{L}$ is the smoothness constant of $J_r(\pi_{\lambda^*(\theta);\theta})$ whose expression is in Lemma 9 in Appendix.*

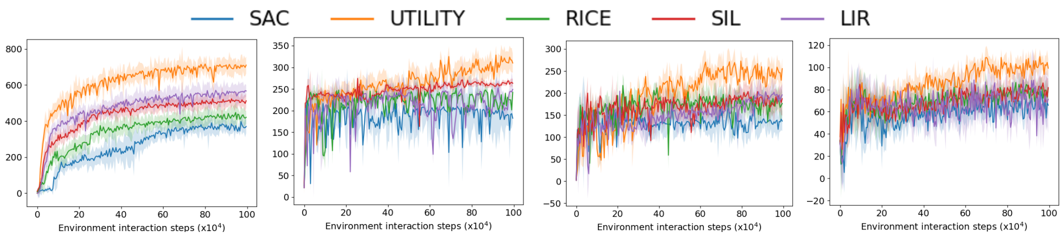
Theorem 4 shows that when the state-action space is finite and r_θ is linear, Algorithm 1 can find an optimal policy asymptotically when the iteration numbers N and $\bar{N}$ go to infinity.

5 EXPERIMENT

This section provides experiment results for the proposed framework. In specific, we aim to answer the question: How does the proposed framework (UTILITY) compare to other RL improvement methods in terms of improving the RL performance. Towards this end, we introduce three RL improvement methods for comparisons. (i) **Fine-tune policy on initial states and critical states (RICE)** (Cheng et al., 2024): This method fine-tunes the policy starting at the original initial states and the states that are most influential to the cumulative reward. Note that (Guo et al., 2021b; Cheng et al., 2023) also use the most influential states to improve performance and (Cheng et al., 2024) shows performance superiority over (Guo et al., 2021b; Cheng et al., 2023), thus we pick (Cheng et al., 2024) to compare. (ii) **Self imitation learning (SIL)** (Oh et al., 2018): This method reproduces previous good decisions in order to encourage deep exploration. (iii) **Learning intrinsic reward (LIR)**: This method aims to learn an intrinsic reward $\tilde{r}$ to formulate the shaping reward $r + \tilde{r}$ (Zheng et al., 2018). We choose these three methods to compare because they respectively belong to three different categories: XRL method (RICE), reward shaping method (LIR), and other methods that can improve RL (SIL). We use soft actor-critic (SAC) (Haarnoja et al., 2018) as the baseline RL algorithm that all the above RL improvement methods use and improve from. We aim to show the

Table 1: Experiment results.

	SAC	UITLITY	RICE	SIL	LIR
Delayed HalfCheetah	383.45 ± 45.50	715.96 ± 42.78	456.14 ± 36.32	510.34 ± 39.28	548.28 ± 47.94
Delayed Hopper	192.90 ± 27.18	317.99 ± 19.62	232.55 ± 16.96	263.46 ± 20.72	247.27 ± 31.93
Delayed Walker2d	134.91 ± 20.80	242.63 ± 14.11	177.45 ± 20.14	172.28 ± 24.57	204.72 ± 25.99
Delayed Ant	68.11 ± 12.52	105.80 ± 14.38	77.01 ± 10.89	81.05 ± 13.43	78.23 ± 13.11



(a) Delayed HalfCheetah (b) Delayed Hopper (c) Delayed Walker (d) Delayed Ant

Figure 2: Improvement curve.

improvement of the above RL improvement methods compared to SAC. We also provide the results of using other baseline RL algorithms in Appendix F.1.

We test the algorithms on delayed MuJoCo environments (Zheng et al., 2018; Memarian et al., 2021; Oh et al., 2018) where the rewards are accumulated for 20 time steps and provided only at the end of these periods. Note that this makes the reward become sparse, and we include the additional experiment results on dense reward in Appendix F.2. We use four different delayed MuJoCo tasks: delayed HalfCheetah, delayed Hopper, delayed Walker2d, and delayed Ant. Following (Finn et al., 2017), each episode has the length of 100 in our experiments. We also provide the results for the case where the episode length is $1,000$ in Appendix F.3.

Figure 2 shows the learning curves of the algorithms where the x-axis is the interaction steps with the MDP environment and the y-axis is the cumulative reward. We plot both the mean (i.e., the solid line) and standard deviation (i.e, the shadow area) of the algorithms. The mean and standard deviation are computed using five random seeds. From the figures, we can observe that UTILITY improves the baseline RL algorithm SAC by a large margin. While the other three methods (i.e., RICE, SIL, and LIR) can also improve SAC to some extent, UTILITY achieves the highest cumulative reward. This is due to the fact that UTILITY (i) learns a shaping reward that makes it easier to learn a good policy and (ii) discourages the policy from making the mistakes (i.e., the "misleading" state-action pairs) made by SAC. Note that the learned shaping reward is dense while the ground truth reward in the delayed MuJoCo environments is sparse, so that UTILITY can guide the learned policy to achieve higher reward. In contrast, RICE and SIL still suffer from the sparse reward. While LIR can also learn a dense shaping reward, UTILITY has the domain knowledge $r - \hat{r}$ formulated by the high-level explanation to help better shape the reward. Moreover, UTILITY has the "misleading" state-action pairs to avoid. Note that the learned shaping reward not only helps in the sparse reward scenario, we include additional results in Appendix F.2 to show that UTILITY can still largely improve SAC when the ground truth reward is dense.

Table 1 shows the final performance of the algorithms. We can observe that UTILITY has the highest cumulative reward among all the algorithms.

The ablation study. Since UTILITY uses both the high-level explanation (i.e., the learned reward) to shape the ground truth reward and the low-level explanation (i.e., misleading state-action pairs) to formulate a constraint to improve SAC, we include an ablation study to separately study the effect of the shaping reward and the constraint. In specific, we consider two methods: "shaping only" and "constraint only". The "shaping only" method only uses the high-level explanation to learn a shaping reward but does not uses the low-level explanation to formulate a constraint. The "constraint only" method only uses the low-level explanation to formulate the constraint but does not shape the original reward. We include the results for the delayed environments in Table 2 and the results for the dense environments in Appendix F.4. The results in Table 2 and Appendix F.4 show that both the “shaping only” and “constraint only” methods can improve SAC. Moreover, the “shaping only” method has a larger impact to improve the performance. This is because the shaping reward improves the policy globally as it changes the reward value for all (s, a), while the constraint may only improve the policy locally around the misleading state-action pairs.

Table 2: Ablation study.

	SAC	UITLITY	shaping only	constraint only
Delayed HalfCheetah	383.45 ± 45.50	715.96 ± 42.78	695.63 ± 33.66	422.15 ± 22.86
Delayed Hopper	192.90 ± 27.18	317.99 ± 19.62	289.10 ± 18.41	210.12 ± 15.77
Delayed Walker2d	134.91 ± 20.80	242.63 ± 14.11	211.37 ± 18.64	175.66 ± 15.27
Delayed Ant	68.11 ± 12.52	105.80 ± 14.38	88.18 ± 8.66	75.16 ± 6.58

Evaluation of the generated two-level explanation. Following (Guo et al., 2021b; Cheng et al., 2023), we use fidelity as the metric to respectively evaluate the high-level and low-level explanations. The fidelity means the correctness of the two-level explanation. Since the two-level explanation is to explain why the RL agent is not optimal, one way to validate the fidelity of the explanation is to see whether the cumulative reward increases after we improve from the explanations.

From the last two columns in Table 2, we can see that both the high-level and low-level explanations are correct explanations because both the shaping only method and the constraint only method improve the performance. Moreover, the shaping only method (the fourth column in Table 2) has a higher cumulative reward than LIR (the last column in Table 1), and the constraint only method (the last column in Table 2) has a higher cumulative reward than RICE (the fourth column in Table 1). This shows the high fidelity of our two-level explanation.

To compare the fidelity of our explanation with other methods, we fix the improvement method and change the explanation to compare. For the low-level explanation, we compare with RICE. In specific, we still use the constraint only method but now the constraint is to discourage from visiting the critical states identified by RICE. We refer to this method as “RICE+constaint”. Note that it is expected that “RICE+constaint” has low fidelity in our case because RICE does not aim to explain why the RL agent is not optimal. For the high-level explanation, since there is no existing XRL method to compare, we use our shaping-only method without the domain knowledge $r - \hat{r}$ to compare. We refer to this method as “shaping without $r - \hat{r}$”. We include the results for sparse reward in Table 3 and the results for dense reward in Appendix F.5. The results show that both the high-level and low-level explanations of UTILITY have high fidelity.

Table 3: Fidelity comparison.

	SAC	shaping only (ours)	shaping without $r - \hat{r}$	constraint only (ours)	RICE+constraint
Delayed HalfCheetah	383.45 ± 45.50	695.63 ± 33.66	611.08 ± 39.44	422.15 ± 22.86	369.14 ± 19.40
Delayed Hopper	192.90 ± 27.18	289.10 ± 18.41	255.18 ± 16.57	210.12 ± 15.77	181.45 ± 12.11
Delayed Walker2d	134.91 ± 20.80	211.37 ± 18.64	191.15 ± 11.26	175.66 ± 15.27	140.26 ± 11.53
Delayed Ant	68.11 ± 12.52	88.18 ± 8.66	77.11 ± 5.52	75.16 ± 6.58	63.11 ± 4.28

Computation time study. While the triple-loop structure of Algorithm 1 looks computationally expensive, in practice, we can significantly accelerate the algorithm using warm start. We elaborate how we use warm start to accelerate Algorithm 1 and provide empirical results to show that Algorithm 1 is not slower than the baselines in Appendix F.6.

6 Conclusion

This paper proposes a theoretical and systematic framework that utilizes XRL to improve RL. We first provide an explanation for why the RL agent is not optimal, and then formulate the problem of utilizing the explanation to improve RL as a constrained bi-level optimization problem. We propose a novel theoretical framework to solve this problem, and use experiments to validate that the proposed framework can improve the RL performance. Despite the benefit, one limitation of the proposed algorithm is that it requires to interact with the environment. Therefore, one future work is to extend our method to the offline RL setting.

REFERENCES

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning*, pp. 1–8, 2004.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International Conference on Machine Learning*, pp. 22–31, 2017.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

Dan Amir and Ofra Amir. Highlights: Summarizing agent behavior to people. In *International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1168–1176, 2018.

Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, 2021.

Akanksha Atrey, Kaleigh Clary, and David Jensen. Exploratory not explanatory: Counterfactual analysis of saliency maps for deep reinforcement learning. In *International Conference on Learning Representations*, 2019.

Osbert Bastani, Yewen Pu, and Armando Solar-Lezama. Verifiable reinforcement learning via policy extraction. *Advances in Neural Information Processing Systems*, 31:2499–2509, 2018.

Amir Beck. *First-order methods in optimization*. Society for Industrial and Applied Mathematics, 2017.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in Neural Information Processing Systems*, pp. 1479–1487, 2016.

Tom Bewley and Jonathan Lawry. Tripletree: A versatile interpretable representation of black box agents and their environments. In *AAAI Conference on Artificial Intelligence*, volume 35, pp. 11415–11422, 2021.

Stephen P Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

Zelei Cheng, Xian Wu, Jiahao Yu, Wenhai Sun, Wenbo Guo, and Xinyu Xing. StateMask: Explaining deep reinforcement learning through state mask. In *Advances in Neural Information Processing Systems*, 2023.

Zelei Cheng, Xian Wu, Jiahao Yu, Sabrina Yang, Gang Wang, and Xinyu Xing. Rice: Breaking through the training bottlenecks of reinforcement learning with explanation. In *International Conference on Machine Learning*, 2024.

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 8103–8112, 2018.

Rati Devidze, Parameswaran Kamalaruban, and Adish Singla. Exploration-guided reward shaping for reinforcement learning under sparse rewards. *Advances in Neural Information Processing Systems*, pp. 5829–5842, 2022.

Sam Michael Devlin and Daniel Kudenko. Dynamic potential-based reward shaping. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 433–440, 2012.

Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135, 2017.

Riccardo Guidotti, Anna Monreale, Fosca Giannotti, Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. Factual and counterfactual explanations for black box decision making. *IEEE Intelligent Systems*, 34(6):14–23, 2019.

Sihang Guo, Ruohan Zhang, Bo Liu, Yifeng Zhu, Dana Ballard, Mary Hayhoe, and Peter Stone. Machine versus human attention in deep reinforcement learning tasks. *Advances in Neural Information Processing Systems*, 34:25370–25385, 2021a.

Wenbo Guo, Xian Wu, Usmann Khan, and Xinyu Xing. Edge: Explaining deep reinforcement learning policies. *Advances in Neural Information Processing Systems*, 34:12222–12236, 2021b.

Abhishek Gupta, Aldo Pacchiano, Yuexiang Zhai, Sham Kakade, and Sergey Levine. Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity. *Advances in Neural Information Processing Systems*, 35:15281–15295, 2022.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pp. 1352–1361, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870, 2018.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Yujing Hu, Weixun Wang, Hangtian Jia, Yixiang Wang, Yingfeng Chen, Jianye Hao, Feng Wu, and Changjie Fan. Learning to utilize shaping rewards: A new approach of reward shaping. *Advances in Neural Information Processing Systems*, 33:15931–15941, 2020.

Sandy H Huang, Kush Bhatia, Pieter Abbeel, and Anca D Dragan. Establishing appropriate trust via critical states. In *IEEE International Conference on Intelligent Robots and Systems*, pp. 3929–3936, 2018.

Alexis Jacq, Johan Ferret, Olivier Pietquin, and Matthieu Geist. Lazy-MDPs: Towards interpretable rl by learning when to act. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 669–677, 2022.

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*, pp. 16291–16325, 2023.

Harshat Kumar, Alec Koppel, and Alejandro Ribeiro. On the sample complexity of actor-critic method for reinforcement learning with function approximation. *Machine Learning*, pp. 1–35, 2023.

Zhengxian Lin, Kin-Ho Lam, and Alan Fern. Contrastive explanations for reinforcement learning via embedded self predictions. In *International Conference on Learning Representations*, 2020.

Shicheng Liu and Minghui Zhu. Distributed inverse constrained reinforcement learning for multi-agent systems. *Advances in Neural Information Processing Systems*, 35:33444–33456, 2022.

Shicheng Liu and Minghui Zhu. Meta inverse constrained reinforcement learning: Convergence guarantee and generalization analysis. In *The Twelfth International Conference on Learning Representations*, 2023.

Shicheng Liu and Minghui Zhu. Learning multi-agent behaviors from distributed and streaming demonstrations. *Advances in Neural Information Processing Systems*, 36, 2024a.

Shicheng Liu and Minghui Zhu. In-trajectory inverse reinforcement learning: Learn incrementally from an ongoing trajectory. *arXiv preprint arXiv:2410.15612*, 2024b.

Yongshuai Liu, Jiaxin Ding, and Xin Liu. IPO: Interior-point policy optimization under constraints. In *AAAI Conference on Artificial Intelligence*, volume 34, pp. 4940–4947, 2020.

Farzan Memarian, Wonjoon Goo, Rudolf Lioutikov, Scott Niekum, and Ufuk Topcu. Self-supervised online reward shaping in sparse-reward environments. In *IEEE International Conference on Intelligent Robots and Systems*, pp. 2369–2375, 2021.

Stephanie Milani, Nicholay Topin, Manuela Veloso, and Fei Fang. Explainable reinforcement learning: A survey and comparative review. *ACM Computing Surveys*, 2023.

Andrew Y Ng, Daishi Harada, and Stuart J Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning*, pp. 278–287, 1999.

Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. In *International conference on machine learning*, pp. 3878–3887, 2018.

Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. Count-based exploration with neural density models. In *International Conference on Machine Learning*, pp. 2721–2730, 2017.

Nikaash Puri, Sukriti Verma, Piyush Gupta, Dhruv Kayastha, Shripad Deshmukh, Balaji Krishnamurthy, and Sameer Singh. Explain your move: Understanding agent actions using specific and relevant feature attribution. In *International Conference on Learning Representations*, 2019.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods. In *International Conference on Machine Learning*, pp. 9133–9143, 2020.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Matthew E Taylor. Improving reinforcement learning with human input. In *International Joint Conference on Artificial Intelligence*, pp. 5724–5728, 2018.

Matthew E Taylor, Nicholas Nissen, Yuan Wang, and Neda Navidi. Improving reinforcement learning with human assistance: An argument for human subject studies with hippo gym. *Neural Computing and Applications*, 35(32):23429–23439, 2023.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. In *International Conference on Learning Representations*, 2018.

Abhinav Verma, Vijayaraghavan Murali, Rishabh Singh, Pushmeet Kohli, and Swarat Chaudhuri. Programmatically interpretable reinforcement learning. In *International Conference on Machine Learning*, pp. 5045–5054, 2018.

George A Vouros. Explainable deep reinforcement learning: State of the art and challenges. *ACM Computing Surveys*, 55(5):1–39, 2022.

Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural policy gradient methods: Global optimality and rates of convergence. In *International Conference on Learning Representations*, 2019.

Zhaodong Wang and Matthew E Taylor. Improving reinforcement learning with confidence-based demonstrations. In *International Joint Conference on Artificial Intelligence*, pp. 3027–3033, 2017.

Yuansheng Xie, Soroush Vosoughi, and Saeed Hassanpour. Towards interpretable deep reinforcement learning models via inverse reinforcement learning. In *International Conference on Pattern Recognition*, pp. 5067–5074, 2022.

Siyuan Xu and Minghui Zhu. Efficient gradient approximation method for constrained bilevel optimization. In *AAAI Conference on Artificial Intelligence*, volume 37, pp. 12509–12517, 2023.

Tengyu Xu, Yingbin Liang, and Guanghui Lan. CRPO: A new approach for safe reinforcement learning with convergence guarantee. In *International Conference on Machine Learning*, pp. 11480–11491, 2021.

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

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

Zeyu Zheng, Junhyuk Oh, and Satinder Singh. On learning intrinsic rewards for policy gradient methods. *Advances in Neural Information Processing Systems*, pp. 4649–4659, 2018.

Zhengyuan Zhou, Michael Bloem, and Nicholas Bambos. Infinite time horizon maximum causal entropy inverse reinforcement learning. *IEEE Transactions on Automatic Control*, 63(9):2787–2802, 2017.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI Conference on Artificial Intelligence*, pp. 1433–1438, 2008.

A Appendix

This appendix has three sections. Section B provides additional content of the two-level explanation. Section C provides notions and notations that serve as the building blocks of the appendix. Section D provides the proof of all the lemmas and theorems in the paper. Section F provides experiment details.

B Two-level explanation

This section has two subsections. Subsection B.1 provides the derivation steps of how we come up with the mathematical metric l to find "misleading" state-action pairs in Section 3 and proves that a state-action pair $(s,a) \in \mathcal{D}$ is misleading if $l(s,a) > 0$. Subsection B.2 provides an algorithm to learn the Q-function $Q_r^{\hat{\pi}_A}$.

B.1 Justification of the mathematical metric to find the "misleading" state-action pairs

Since the misleading state-action pairs lead the policy π_A to be non-optimal, we can say that the policy π_A will be an optimal policy if it does not visit misleading state-action pairs. Therefore, in order to identify "misleading" state-action pairs using Q-functions, we need to first build a connection between Q-functions and policy:

Definition 2. *(i) When we say that a Q-function $\bar{Q}_r^\pi$* ***indicates*** *a policy π, it means that $\pi(s) = \arg\max_a \bar{Q}_r^\pi(s,a)$ for any $s \in \mathcal{S}$ where $\pi(s)$ is the set of actions that the policy π will choose at the state s.*
(ii) We use $\bar{\mathcal{Q}}_r^\pi$ to denote the set of all the Q-functions that indicate the policy π, and thus we can say that $\bar{\mathcal{Q}}_r^\pi$ ***indicates*** *the policy π. Moreover, when we say that $\bar{\mathcal{Q}}_r^\pi$* ***indicates*** *an action a at a state s, it means that $a \in \arg\max_{a'} \bar{Q}_r^\pi(s,a')$ where $\bar{Q}_r^\pi$ is an arbitrary Q-function in $\bar{\mathcal{Q}}_r^\pi$.*
(iii) When we say that Q_r^π ***disagrees with*** *$\bar{\mathcal{Q}}_r^\pi$* ***on*** *(s,a), it means that $\bar{\mathcal{Q}}_r^\pi$ indicates the action a at the state s but $a \notin \arg\max_{a'} Q_r^\pi(s,a')$.*

Definition 2 establishes a connection between Q-functions and policy. With this connection, the following theorem provides a way to define a mathematical metric to find "misleading" state-action pairs using Q-functions:

Theorem 5. *The policy π_A will be an optimal policy if π_A never visits any state-action pair (s,a), on which $Q_r^{\pi_A}$ disagrees with $\bar{\mathcal{Q}}_r^{\pi_A}$.*

Proof. Suppose there is no state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$, on which $Q_r^{\pi_A}$ disagrees with $\bar{\mathcal{Q}}_r^{\pi_A}$. Then for any $(s,a) \in \mathcal{S} \times \mathcal{A}$, if $a \in \arg\max_{a'} \bar{Q}_r^{\pi_A}(s,a')$ where $\bar{Q}_r^{\pi_A} \in \bar{\mathcal{Q}}_r^{\pi_A}$, it holds that $a \in \arg\max_{a'} Q_r^{\pi_A}(s,a')$. Since $\pi_A(s) = \arg\max_a \bar{Q}_r^{\pi_A}(s,a)$ where $\bar{Q}_r^{\pi_A} \in \bar{\mathcal{Q}}_r^{\pi_A}$, it holds that $\pi_A(s) \subseteq \arg\max_a Q_r^{\pi_A}(s,a)$ for any $s \in \mathcal{S}$. Recall that $V_r^{\pi_A}$ and $Q_r^{\pi_A}$ are respectively the value function and Q-function of the policy π_A under the reward function r, then it holds that $Q_r^{\pi_A}(s,a) = r(s,a) + \gamma E_{s'\sim P(\cdot|s,a)}[V_r^{\pi_A}(s')]$ and $V_r^{\pi_A}(s) = E_{a'\sim\pi_A(\cdot|s)}[Q_r^{\pi_A}(s,a')]$. Since $\pi_A(s) \subseteq \arg\max_a Q_r^{\pi_A}(s,a)$ for any $s \in \mathcal{S}$, then $V_r^{\pi_A}(s) = E_{a'\sim\pi_A(\cdot|s)}[Q_r^{\pi_A}(s,a')] = \max_{a'} Q_r^{\pi_A}(s,a')$ for any $s \in \mathcal{S}$. Therefore, the Q-function $Q_r^{\pi_A}$ satisfies the Bellman optimality equation $Q_r^{\pi_A}(s,a) = r(s,a) + \gamma E_{s'\sim P(s'|s,a)}[\max_{a'} Q_r^{\pi_A}(s',a')]$ for any $(s,a) \in \mathcal{S}\times\mathcal{A}$, and thus $Q_r^{\pi_A}$ is the optimal Q-function because the Bellman optimality equation is uniquely satisfied by the optimal Q-function. Since $\pi_A(s) \subseteq \arg\max_a Q_r^{\pi_A}(s,a)$ for any $s \in \mathcal{S}$, the policy π_A should be an optimal policy. $\square$

Since the misleading state-action pairs lead the policy π_A to be non-optimal, we can say that the policy π_A will be an optimal policy if it has zero probability to visit misleading state-action pairs. Therefore, Theorem 5 shows that the "misleading" state-action pairs can be mathematically defined as the state-action pairs, on which $Q_r^{\pi_A}$ disagrees with $\bar{\mathcal{Q}}_r^{\pi_A}$. Therefore, we can develop a mathematical metric to find the top K "misleading" state-action pairs in the demonstration set $\mathcal{D}$ using Q-functions. Since the demonstration set $\mathcal{D}$ is generated by the policy π_A and $\bar{\mathcal{Q}}_r^{\pi_A}$ indicates the policy π_A, $\bar{\mathcal{Q}}_r^{\pi_A}$ will indicate the action a at the state s for any $(s,a) \in \mathcal{D}$. In order to find the "misleading" state-actions, we need to find $(s,a) \in \mathcal{D}$ such that $a \notin \arg\max_{a'} Q_r^{\pi_A}(s,a')$ or in other words, $Q_r^{\pi_A}(s,a) < \max_{a'} Q_r^{\pi_A}(s,a')$. Therefore, we can use the function $l(s,a) \triangleq \max_{a'} Q_r^{\pi_A}(s,a') - Q_r^{\pi_A}(s,a)$ as the metric to identify "misleading" state-action pairs in the demonstration set $\mathcal{D}$. The larger the loss $l(s,a)$ is, the more "misleading" the state-action pair (s,a) is, because the Q value of the chosen action a has a larger gap from the maximum Q value at the state s under the Q-function $Q_r^{\pi_A}$.

B.2 Method

In this subsection, we use a standard regression method to learn the Q-function $Q_r^{\hat{\pi}_A}$. In specific, we roll out the policy $\hat{\pi}_A$ to generate a set $\bar{\mathcal{D}}$ of many (s,a) samples. For each $(s,a) \in \mathcal{D}$, we use $\hat{\pi}_A$ to generate many trajectories starting from (s,a) and use these trajectories to estimate $Q_r^{\hat{\pi}_A}(s,a)$. Since we can generate many trajectories, we can estimate the Q value $Q_r^{\hat{\pi}_A}(s,a)$ for each $(s,a) \in \bar{\mathcal{D}}$ quite accurately. Then we use a neural network Q_ω parameterized by ω to solve the following regression problem:

$$\min_\omega \sum_{(s,a)\in\mathcal{D}} ||Q_\omega(s,a) - Q_r^{\hat{\pi}_A}(s,a)||^2.$$

C Notions and notations

The shaping reward function $r_\theta(r(s,a), r(s,a)-\hat{r}(s,a))$ is a function of $r(s,a)$ and $r(s,a)-\hat{r}(s,a)$, and $r(s,a)$ and $r(s,a)-\hat{r}(s,a)$ are both functions of (s,a). Therefore, the shaping reward function r_θ is also a function of (s,a). For simple notations, we use $r_\theta(s,a)$ to denote the shaping reward of (s,a). Given the policy π and the parameters (λ,θ)s, the corresponding constrained soft Q-function and constrained soft value function are:

$$Q^\pi_{\lambda;\theta}(s,a) \triangleq r_\theta(s,a) - \lambda c(s,a) + \gamma \int_{s'\in\mathcal{S}} P(s'|s,a) V^\pi_{\lambda;\theta}(s') ds',$$

$$V^\pi_{\lambda;\theta}(s) \triangleq E^\pi\left[\sum_{t=0}^\infty \gamma^t (r_\theta(s_t,a_t) - \lambda c(s_t,a_t) - \log\pi(a_t|s_t)) \middle| s_0 = s\right].$$

Moreover, it can be shown (Zeng et al., 2022; Haarnoja et al., 2017; Liu & Zhu, 2024b) that $\exp(V^\pi_{\lambda;\theta}(s)) = \int_{a\in\mathcal{A}} \exp(Q^\pi_{\lambda;\theta}(s,a)) da$.

The constrained soft policy (Liu & Zhu, 2022; 2023) is

$$\pi_{\lambda;\theta}(a|s) = \frac{\exp(Q_{\lambda;\theta}(s,a))}{\exp(V_{\lambda;\theta}(s))}, \tag{3}$$

$$Q_{\lambda;\theta}(s,a) = r_\theta(s,a) - \lambda c(s,a) + \gamma \int_{s'\in\mathcal{S}} P(s'|s,a) V_{\lambda;\theta}(s') ds', \tag{4}$$

$$V_{\lambda;\theta}(s) = \log \int_{a\in\mathcal{A}} \exp(Q_{\lambda;\theta}(s,a))da. \tag{5}$$

We can obtain the constrained soft policy via soft Q-learning (Haarnoja et al., 2017) or soft actor-critic (Haarnoja et al., 2018) by treating $r_\theta - \lambda c$ as the new reward function. We define the cumulative cost under the policy π starting from (s,a) as $Q_c^\pi(s,a) \triangleq E^\pi[\sum_{t=0}^\infty \gamma^t c(s_t,a_t)|s_0 = s, a_0 = a]$ and the cumulative cost starting from s as $V_c^\pi(s) \triangleq E^\pi[\sum_{t=0}^\infty \gamma^t c(s_t,a_t)|s_0 = s]$. We define the cumulative reward under the policy π starting from (s,a) as $Q_r^\pi(s,a) \triangleq E^\pi[\sum_{t=0}^\infty \gamma^t r(s_t,a_t)|s_0 = s, a_0 = a]$ and the cumulative reward starting from s as $V_r^\pi(s) \triangleq E^\pi[\sum_{t=0}^\infty \gamma^t r(s_t,a_t)|s_0 = s]$. We define the cumulative reward gradient under the policy π starting from (s,a) as $Q^\pi_{\nabla_\theta r_\theta}(s,a) \triangleq E^\pi[\sum_{t=0}^\infty \gamma^t \nabla_\theta r_\theta(s_t,a_t)|s_0 = s, a_0 = a]$ and the cumulative reward starting from s as $V^\pi_{\nabla_\theta r_\theta}(s) \triangleq E^\pi[\sum_{t=0}^\infty \gamma^t \nabla_\theta r_\theta(s_t,a_t)|s_0 = s]$. We define the state visitation frequency under a policy π as $\psi^\pi(s) \triangleq E^\pi[\sum_{t=0}^\pi \gamma^t \mathbb{1}\{s_t = s\}]$ and state-action visitation frequency as $\psi^\pi(s,a) \triangleq E^\pi[\sum_{t=0}^\pi \gamma^t \mathbb{1}\{s_t = s, a_t = a\}]$ where $\mathbb{1}\{\cdot\}$ is the indicator function.

D Proof

This section provides the proof of all the lemmas and theorems in the paper.

Lemma 5. *The gradients of the constrained soft policy are respectively* $\nabla_\lambda \log \pi_{\lambda;\theta}(a|s) = V_c^{\pi_{\lambda;\theta}}(s) - Q_c^{\pi_{\lambda;\theta}}(s,a)$ *and* $\nabla_\theta \log \pi_{\lambda;\theta}(a|s) = Q^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s,a) - V^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s)$.

Proof. Recall from (3), we know that $\nabla_\lambda \log \pi_{\lambda;\theta}(a|s) = \nabla_\lambda Q_{\lambda;\theta}(s,a) - \nabla_\lambda V_{\lambda;\theta}(s)$. Recall from (4), we know that

$$\begin{aligned}
&\nabla_\lambda Q_{\lambda;\theta}(s,a) = -c(s,a) + \gamma \int_{s'\in\mathcal{S}} P(s'|s,a)\nabla_\lambda V_{\lambda;\theta}(s')ds',\\
&\overset{(a)}{=} -c(s,a) + \gamma \int_{s'\in\mathcal{S}} P(s'|s,a)\frac{1}{\exp(V_{\lambda;\theta}(s'))}\int_{a'\in\mathcal{A}} \nabla_\lambda \exp(Q_{\lambda;\theta}(s',a'))da'ds',\\
&= -c(s,a) + \gamma \int_{s'\in\mathcal{S}}\int_{a'\in\mathcal{A}} P(s'|s,a)\frac{\exp(Q_{\lambda;\theta}(s',a'))}{\exp(V_{\lambda;\theta}(s'))}\nabla_\lambda Q_{\lambda;\theta}(s',a')da'ds',,\\
&\overset{(b)}{=} -c(s,a) + \gamma \int_{s'\in\mathcal{S}}\int_{a'\in\mathcal{A}} P(s'|s,a)\pi_{\lambda;\theta}(a|s)\nabla_\lambda Q_{\lambda;\theta}(s',a')da'ds',\\
&= -c(s,a) + \gamma \int_{s'\in\mathcal{S}}\int_{a'\in\mathcal{A}} P(s'|s,a)\pi_{\lambda;\theta}(a|s)\left[-c(s',a') + \gamma\int_{s''\in\mathcal{S}} P(s''|s',a')\nabla_\lambda V_{\lambda;\theta}(s'')ds''\right],
\end{aligned}$$

where (a) follows (5), (b) follows (3). Keep the expansion, we can see that

$$\nabla_\lambda Q_{\lambda;\theta}(s,a) = -E^{\pi_{\lambda;\theta}}[\sum_{t=0}^\infty \gamma^t c(s_t,a_t)|s_0 = s, a_0 = a] = -Q_c^{\pi_{\lambda;\theta}}(s,a). \tag{6}$$

Similarly, we can get that:

$$\nabla_\lambda V_{\lambda;\theta}(s) = -E^{\pi_{\lambda;\theta}}[\sum_{t=0}^\infty \gamma^t c(s_t,a_t)|s_0 = s] = -V_c^{\pi_{\lambda;\theta}}(s), \tag{7}$$

$$\nabla_\theta Q_{\lambda;\theta}(s,a) = E^{\pi_{\lambda;\theta}}[\sum_{t=0}^\infty \gamma^t \nabla_\theta r_\theta(s_t,a_t)|s_0 = s, a_0 = a] = Q^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s,a), \tag{8}$$

$$\nabla_\theta V_{\lambda;\theta}(s) = E^{\pi_{\lambda;\theta}}[\sum_{t=0}^\infty \gamma^t \nabla_\theta r_\theta(s_t,a_t)|s_0 = s] = V^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s), \tag{9}$$

Therefore, we can compute the gradients

$$\begin{aligned}
\nabla_\lambda \log \pi_{\lambda;\theta}(a|s) &= \nabla_\lambda Q_{\lambda;\theta}(s,a) - \nabla_\lambda V_{\lambda;\theta}(s) = V_c^{\pi_{\lambda;\theta}}(s) - Q_c^{\pi_{\lambda;\theta}}(s,a),\\
\nabla_\theta \log \pi_{\lambda;\theta}(a|s) &= \nabla_\theta Q_{\lambda;\theta}(s,a) - \nabla_\theta V_{\lambda;\theta}(s) = Q^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s,a) - V^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s).
\end{aligned}$$

$\square$

Lemma 6. *The constrained soft operator $\mathcal{T}^{soft}_{\lambda;\theta}$:*

$$(\mathcal{T}^{soft}_{\lambda;\theta} Q)(s,a) \triangleq r_\theta(s,a) - \lambda c(s,a) + \gamma \int_{s' \in \mathcal{S}} P(s'|s,a) \log\left[\int_{a' \in \mathcal{A}} \exp(Q(s',a'))da'\right] ds',$$
$$(\mathcal{T}^{soft}_{\lambda;\theta} V)(s) \triangleq \log\left[\int_{a \in \mathcal{A}} \exp\left(r_\theta(s,a) - \lambda c(s,a) + \gamma \int_{s' \in \mathcal{S}} P(s'|s,a) V(s') ds'\right) da\right],$$

is a contraction map with constant γ.

Proof. It has been proved that $\mathcal{T}^{\text{soft}}_{\lambda;\theta} Q$ is a contraction map with constant γ (Appendix A.2 in (Haarnoja et al., 2017) if we replace r with $r_\theta - \lambda c$). Here we show that $\mathcal{T}^{\text{soft}}_{\lambda;\theta} V$ is a contraction map with constant γ. Define a norm of V as $||V_1 - V_2|| = \sup_{s \in \mathcal{S}} |V_1(s) - V_2(s)|$ and suppose $||V_1 - V_2|| = \epsilon$. Then we have that

$$\begin{aligned}
\mathcal{T}^{\text{soft}}_{\lambda;\theta} V_1(s) &= \log\left[\int_{a \in \mathcal{A}} \exp\left(r_\theta(s,a) - \lambda c(s,a) + \gamma \int_{s' \in \mathcal{S}} P(s'|s,a) V_1(s') ds'\right) da\right], \\
&\leq \log\left[\int_{a \in \mathcal{A}} \exp\left(r_\theta(s,a) - \lambda c(s,a) + \gamma \int_{s' \in \mathcal{S}} P(s'|s,a) [V_2(s') + \epsilon] ds'\right) da\right], \\
&= \log\left[\int_{a \in \mathcal{A}} \exp\left(r_\theta(s,a) - \lambda c(s,a) + \gamma \int_{s' \in \mathcal{S}} P(s'|s,a) V_2(s') ds' + \gamma\epsilon\right) da\right], \\
&= \log\left[\int_{a \in \mathcal{A}} \exp(\gamma\epsilon) \exp\left(r_\theta(s,a) - \lambda c(s,a) + \gamma \int_{s' \in \mathcal{S}} P(s'|s,a) V_2(s') ds'\right) da\right], \\
&= \mathcal{T}^{\text{soft}}_{\lambda;\theta} V_2(s) + \gamma\epsilon.
\end{aligned}$$

Similarly, we can get $\mathcal{T}^{\text{soft}}_{\lambda;\theta} V_1(s) \geq \mathcal{T}^{\text{soft}}_{\lambda;\theta} V_2(s) - \gamma\epsilon$. Therefore, $||\mathcal{T}^{\text{soft}}_{\lambda;\theta} V_1 - \mathcal{T}^{\text{soft}}_{\lambda;\theta} V_2|| \leq \gamma\epsilon = \gamma||V_1 - V_2||$. □

Lemma 7. *It holds that $Q^{\pi_{\tilde{n}+1}}_{\lambda;\theta}(s,a) \geq \mathcal{T}^{soft}_{\lambda;\theta}(Q^{\pi_{\tilde{n}}}_{\lambda;\theta})(s,a)$ and $V^{\pi_{\tilde{n}+1}}_{\lambda;\theta}(s) \geq \mathcal{T}^{soft}_{\lambda;\theta}(V^{\pi_{\tilde{n}}}_{\lambda;\theta})(s)$ for any (s,a).*

Proof.

$$\begin{aligned}
&Q^{\pi_{\tilde{n}+1}}_{\lambda;\theta}(s,a) \overset{(a)}{=} r_\theta(s,a) - \lambda c(s,a) + \gamma \int_{s' \in \mathcal{S}} P(s'|s,a) E_{a' \sim \pi_{\tilde{n}+1}}[Q^{\pi_{\tilde{n}+1}}_{\lambda;\theta}(s',a') - \log \pi_{\tilde{n}+1}(a'|s')] ds', \\
&\overset{(b)}{\geq} r_\theta(s,a) - \lambda c(s,a) + \gamma \int_{s' \in \mathcal{S}} P(s'|s,a) E_{a' \sim \pi_{\tilde{n}+1}(\cdot|s')}[Q^{\pi_{\tilde{n}}}_{\lambda;\theta}(s',a') - \log \pi_{\tilde{n}+1}(a'|s')] ds', \\
&= r_\theta(s,a) - \lambda c(s,a) + \gamma \int_{s' \in \mathcal{S}} P(s'|s,a) \log\left[\int_{a' \in \mathcal{A}} \exp(Q^{\pi_{\tilde{n}}}_{\lambda;\theta}(s',a')) da'\right] ds', \\
&= \mathcal{T}^{\text{soft}}_{\lambda;\theta}(Q^{\pi_{\tilde{n}}}_{\lambda;\theta})(s,a),
\end{aligned}$$

where (a) follow equations (2)-(3) in (Haarnoja et al., 2018) and (b) follows policy improvement theorem (Theorem 4 in (Haarnoja et al., 2017)). Similarly, we can get that

$$\begin{aligned}
V^{\pi_{\tilde{n}+1}}_{\lambda;\theta}(s) &= E_{a \sim \pi_{\tilde{n}+1}(\cdot|s)}[Q^{\pi_{\tilde{n}+1}}_{\lambda;\theta}(s,a) - \log \pi_{\tilde{n}+1}(a|s)], \\
&\geq E_{a \sim \pi_{\tilde{n}+1}(\cdot|s)}[Q^{\pi_{\tilde{n}}}_{\lambda;\theta}(s,a) - \log \pi_{\tilde{n}+1}(a|s)], \\
&= \log\left[\int_{a \in \mathcal{A}} \exp(Q^{\pi_{\tilde{n}}}_{\lambda;\theta}(s,a)) da\right], \\
&= \log\left[\int_{a \in \mathcal{A}} \exp\left(r_\theta(s,a) - \lambda c(s,a) + \gamma \int_{s' \in \mathcal{S}} P(s'|s,a) V^{\pi_{\tilde{n}}}_{\lambda;\theta}(s') ds'\right) da\right], \\
&= \mathcal{T}^{\text{soft}}_{\lambda;\theta}(V^{\pi_{\tilde{n}}}_{\lambda;\theta})(s).
\end{aligned}$$

□

D.1 PROOF OF LEMMA 1

It has been proved in Theorem 1 in (Haarnoja et al., 2017) that the constrained soft policy $\pi_{\lambda;\theta} = \arg\max_\pi J_{r_\theta}(\pi) - \lambda J_c(\pi) + H(\pi)$ where we treat $r_\theta - \lambda c$ as the new reward function. Recall that the dual function is $G(\lambda;\theta) = \max_\pi J_{r_\theta}(\pi) - \lambda(J_c(\pi) - b) + H(\pi)$, therefore, we know that $G(\lambda;\theta) = J_{r_\theta}(\pi_{\lambda;\theta}) - \lambda(J_c(\pi_{\lambda;\theta}) - b) + H(\pi_{\lambda;\theta})$. Since $\pi_{\lambda;\theta}$ is the optimal solution and the G is differentiable to the policy, we know that $\frac{\partial G(\lambda;\theta)}{\partial \pi_{\lambda;\theta}(a|s)} = 0$ for every $(s,a) \in \mathcal{S} \times \mathcal{A}$. Note that

$$\frac{\partial G(\lambda;\theta)}{\partial \pi_{\lambda;\theta}(a|s)} = \frac{\partial}{\partial \pi_{\lambda;\theta}(a|s)} \psi^{\pi_{\lambda;\theta}}(s,a)\Big[r_\theta(s,a) - \lambda c(s,a) - \log \pi_{\lambda;\theta}(a|s)\Big] = 0 \tag{10}$$

$$\begin{aligned}
&\nabla_\lambda G(\lambda;\theta) = \nabla_\lambda \Big[J_{r_\theta}(\pi_{\lambda;\theta}) - \lambda J_c(\pi_{\lambda;\theta}) + H(\pi_{\lambda;\theta})\Big] - (J_c(\pi_{\lambda;\theta}) - b),\\
&= \int_{s\in\mathcal{S}}\int_{a\in\mathcal{A}} \nabla_\lambda \Big\{\psi^{\pi_{\lambda;\theta}}(s,a)\Big[r_\theta(s,a) - \lambda c(s,a) - \log\pi_{\lambda;\theta}(a|s)\Big]\Big\} da ds - (J_c(\pi_{\lambda;\theta}) - b),\\
&= \int_{s\in\mathcal{S}}\int_{a\in\mathcal{A}} \nabla_\lambda \pi_{\lambda;\theta}(a|s) \cdot \frac{\partial}{\partial \pi_{\lambda;\theta}(a|s)} \Big\{\psi^{\pi_{\lambda;\theta}}(s,a)\Big[r_\theta(s,a) - \lambda c(s,a) - \log\pi_{\lambda;\theta}(a|s)\Big]\Big\} da ds\\
&- (J_c(\pi_{\lambda;\theta}) - b),\\
&\overset{(a)}{=} b - J_c(\pi_{\lambda;\theta}),
\end{aligned}$$

where (a) follows (10).

D.2 PROOF OF THEOREM 1 AND THEOREM 2

We first prove Theorem 2 and then prove Theorem 1. The fundamental logic is that we first prove that the dual problem has a unique optimal solution $\lambda^*(\theta)$ and then we prove that the primal problem (i.e. the lower-level problem in (2)) and the dual problem have the same set of optimal solutions. Since the optimal solution of the dual problem is unique, then the optimal solution of the primal problem is also unique.

To show the optimal solution of the dual problem is unique, we prove that the dual function is strictly convex by showing that the Hessian of the dual function to λ is positive definite.

From Lemma 1, we know that $\nabla_\lambda G(\lambda;\theta) = b - J_c(\pi_{\lambda;\theta})$, therefore, we have that

$$\begin{aligned}
&\nabla^2_{\lambda\lambda} G(\lambda;\theta) = -\nabla_\lambda J_c(\pi_{\lambda;\theta}),\\
&= -\nabla_\lambda \int_{s_0\in\mathcal{S}} P_0(s_0) \int_{a_0\in\mathcal{A}} \pi_{\lambda;\theta}(a_0|s_0)\Big[c(s_0,a_0) + \gamma\int_{s_1\in\mathcal{S}} P(s_1|s_0,a_0) Q_c^{\pi_{\lambda;\theta}}(s_1) ds_1\Big] da_0 ds_0,\\
&= -\int_{s_0\in\mathcal{S}} P_0(s_0) \int_{a_0\in\mathcal{A}} \Big\{\nabla_\lambda \pi_{\lambda;\theta}(a_0|s_0) \cdot \Big[c(s_0,a_0) + \gamma \int_{s_1\in\mathcal{S}} P(s_1|s_0,a_0) Q_c^{\pi_{\lambda;\theta}}(s_1) ds_1\Big]\\
&+ \pi_{\lambda;\theta}(a_0|s_0) \cdot \Big[\gamma \int_{s_1\in\mathcal{S}} P(s_1|s_0,a_0) \nabla_\lambda Q_c^{\pi_{\lambda;\theta}}(s_1) ds_1\Big]\Big\} da_0 ds_0,\\
&= -\int_{s_0\in\mathcal{S}} P_0(s_0) \int_{a_0\in\mathcal{A}} \Big\{\nabla_\lambda \pi_{\lambda;\theta}(a_0|s_0) \cdot Q_c^{\pi_{\lambda;\theta}}(s_0,a_0)\\
&+ \pi_{\lambda;\theta}(a_0|s_0) \cdot \gamma \int_{s_1\in\mathcal{S}} P(s_1|s_0,a_0) \nabla_\lambda \Big[\int_{a_1\in\mathcal{A}} \pi_{\lambda;\theta}(a_1|s_1) \cdot Q_c^{\pi_{\lambda;\theta}}(s_1,a_1) da_1 ds_1\Big]\Big\} da_0 ds_0.
\end{aligned}$$

Keep the expansion, we can get that

$$\begin{aligned}
\nabla^2_{\lambda\lambda} G(\lambda;\theta) &= -\int_{s\in\mathcal{S}}\int_{a\in\mathcal{A}} \psi^{\pi_{\lambda;\theta}}(s) \nabla_\lambda \pi_{\lambda;\theta}(a|s) \cdot Q_c^{\pi_{\lambda;\theta}}(s,a) da ds,\\
&= -\int_{s\in\mathcal{S}}\int_{a\in\mathcal{A}} \psi^{\pi_{\lambda;\theta}}(s) \pi_{\lambda;\theta}(a|s) \nabla_\lambda \log \pi_{\lambda;\theta}(a|s) \cdot Q_c^{\pi_{\lambda;\theta}}(s,a) da ds,
\end{aligned}$$

$$\overset{(a)}{=} \int_{s\in\mathcal{S}} \psi^{\pi_{\lambda;\theta}}(s) \int_{a\in\mathcal{A}} \pi_{\lambda;\theta}(a|s)\Big[Q_c^{\pi_{\lambda;\theta}}(s,a) - V_c^{\pi_{\lambda;\theta}}(s)\Big]Q_c^{\pi_{\lambda;\theta}}(s,a)dads, \tag{11}$$

where (a) follows Lemma 5. Note that $V_c^{\pi_{\lambda;\theta}}(s) = E_{a\sim\pi_{\lambda;\theta}(\cdot|s)}[Q_c^{\pi_{\lambda;\theta}}(s,a)]$. If we use the random variable X_{sa} to denote $Q_c^{\pi_{\lambda;\theta}}(s,a)$, then its expectation is $E_{a\sim\pi_{\lambda;\theta}(\cdot|s)}(X_{sa}) = V_c^{\pi_{\lambda;\theta}}(s)$. We know that the variance $Var(X_{sa}) = E[X_{sa}[X_{sa} - E(X_{sa})]]$. Therefore, we can see that the equation (11) is actually a variance:

$$\nabla^2_{\lambda\lambda}G(\lambda;\theta) = \int_{s\in\mathcal{S}} \psi^{\pi_{\lambda;\theta}}(s)Var(X_{sa})ds.$$

From the expression (3), we know that $\pi_{\lambda;\theta}$ is always stochastic. Therefore, the variance $Var(X_{sa}) > 0$. Then we know that $\nabla^2_{\lambda\lambda}G(\lambda;\theta) > 0$ and thus G is strictly convex. Therefore, the optimal solution $\lambda^*(\theta)$ is unique.

Since $G(\lambda;\theta)$ attains its minimum at $\lambda^*(\theta)$, the gradient of G at $\lambda^*(\theta)$ should be zero, i.e., $J_c(\pi_{\lambda^*(\theta);\theta}) - b = 0$. Let p^* and d^* be the optimal value of the primal problem and the dual problem. Since $G(\lambda;\theta) = \max_\pi J_{r_\theta}(\pi) + H(\pi) - \lambda(J_c(\pi) - b)$, we know that $G(\lambda;\theta) \geq J_{r_\theta}(\pi) + H(\pi)$ for any (λ,θ), which means that $d^* \geq p^*$. Therefore, we have that:

$$p^* \leq d^* = G(\lambda^*(\theta);\theta) \overset{(b)}{=} J_{r_\theta}(\pi_{\lambda^*(\theta);\theta}) + H(\pi_{\lambda^*(\theta);\theta}) \leq p^*,$$

where (b) follows the fact that $J_c(\pi_{\lambda^*(\theta);\theta}) - b = 0$. Therefore, $\pi_{\lambda^*(\theta);\theta}$ is an optimal solution of the primal problem. Suppose the primal problem has another optimal solution π', then it holds that $\pi' \in \arg\max_\pi G(\lambda^*(\theta);\theta)$. However, it has been proved in Lemma 1 in (Zhou et al., 2017) that given an arbitrary λ, the optimal policy of $\max_\pi J_{r_\theta}(\pi) + H(\pi) - \lambda(J_c(\pi) - b)$ is unique. Therefore, $\pi_{\lambda^*(\theta);\theta}$ is the unique optimal solution of the primal problem (i.e., the lower-level problem in (2)).

D.3 Proof of Lemma 2

Since $\lambda^*(\theta) = \arg\min G(\lambda;\theta)$, we know that $\nabla_\lambda G(\lambda^*(\theta);\theta) = 0$. Therefore, we have that

$$\begin{aligned}
&\frac{d\nabla_\lambda G(\lambda^*(\theta);\theta)}{d\theta} = 0, \\
&\Rightarrow \nabla^2_{\theta\lambda}G(\lambda^*(\theta);\theta) + \nabla^2_{\lambda\lambda}G(\lambda^*(\theta);\theta)\nabla\lambda^*(\theta) = 0, \\
&\Rightarrow \nabla\lambda^*(\theta) = -[\nabla^2_{\lambda\lambda}G(\lambda^*(\theta);\theta)]^{-1}\nabla^2_{\theta\lambda}G(\lambda^*(\theta);\theta).
\end{aligned} \tag{12}$$

Now we take a look at the term $\nabla^2_{\theta\lambda}G(\lambda;\theta)$. From Lemma 1, we know that $\nabla_\lambda G(\lambda;\theta) = b - J_c(\pi_{\lambda;\theta})$, therefore, we have that

$$\begin{aligned}
&\nabla^2_{\lambda\theta}G(\lambda;\theta) = -\nabla_\theta J_c(\pi_{\lambda;\theta}), \\
&= -\nabla_\theta \int_{s_0\in\mathcal{S}} P_0(s_0) \int_{a_0\in\mathcal{A}} \pi_{\omega;\theta}(a_0|s_0)Q_c^{\pi_{\lambda;\theta}}(s_0,a_0)da_0ds_0, \\
&= -\int_{s_0\in\mathcal{S}} P_0(s_0) \int_{a_0\in\mathcal{A}} \pi_{\omega;\theta}(a_0|s_0)\Big[\nabla_\theta \log \pi_{\omega;\theta}(a_0|s_0) \cdot Q_c^{\pi_{\lambda;\theta}}(s_0,a_0) + \nabla_\theta Q_c^{\pi_{\lambda;\theta}}(s_0,a_0)\Big]da_0ds_0, \\
&= -\int_{s_0\in\mathcal{S}} P_0(s_0) \int_{a_0\in\mathcal{A}} \pi_{\omega;\theta}(a_0|s_0)\Big[\nabla_\theta \log \pi_{\omega;\theta}(a_0|s_0) \cdot Q_c^{\pi_{\lambda;\theta}}(s_0,a_0) \\
&- \nabla_\theta[c(s_0,a_0) + \gamma\int_{s_1\in\mathcal{S}} P(s_1|s_0,a_0)Q_c^{\pi_{\lambda;\theta}}(s_1)ds_1]\Big]da_0ds_0, \\
&= -\int_{s_0\in\mathcal{S}} P_0(s_0) \int_{a_0\in\mathcal{A}} \pi_{\omega;\theta}(a_0|s_0)\Big[\nabla_\theta \log \pi_{\omega;\theta}(a_0|s_0) \cdot Q_c^{\pi_{\lambda;\theta}}(s_0,a_0) \\
&- \gamma\int_{s_1\in\mathcal{S}} P(s_1|s_0,a_0)\nabla_\theta \int_{a_1\in\mathcal{A}} \pi_{\lambda;\theta}(a_1|s_1)Q_c^{\pi_{\lambda;\theta}}(s_1,a_1)da_1ds_1]\Big]da_0ds_0.
\end{aligned}$$

Keep the expansion, we can get that

$$\nabla^2_{\lambda\theta}G(\lambda;\theta) = -\int_{s\in\mathcal{S}}\int_{a\in\mathcal{A}} \psi^{\pi_{\lambda;\theta}}(s,a)\nabla_\theta \log \pi_{\lambda;\theta}(a|s)Q_c^{\pi_{\lambda;\theta}}(s,a)dads,$$

$$\overset{(a)}{=} -\int_{s\in\mathcal{S}}\int_{a\in\mathcal{A}} \psi^{\pi_{\lambda;\theta}}(s,a)\left[Q^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s,a) - V^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s)\right] Q_c^{\pi_{\lambda;\theta}}(s,a)dads, \tag{13}$$

where (a) follows Lemma 5.

Now, we take the full gradient of $\log \pi_{\lambda^*(\theta);\theta}(a|s)$ to θ:

$$\frac{d\log \pi_{\lambda^*(\theta);\theta}(a|s)}{d\theta} = \nabla_\theta \log \pi_{\lambda^*(\theta);\theta}(a|s) + \nabla_\lambda \log \pi_{\lambda^*(\theta);\theta}(a|s)\cdot \nabla\lambda^*(\theta),$$
$$\overset{(b)}{=} Q^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s,a) - V^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s) + (Q_c^{\pi_{\lambda^*(\theta);\theta}}(s,a) - V_c^{\pi_{\lambda^*(\theta);\theta}}(s))[\nabla^2_{\lambda\lambda}G(\lambda^*(\theta);\theta)]^{-1}\nabla^2_{\theta\lambda}G(\lambda^*(\theta);\theta),$$
$$\overset{(c)}{=} Q^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s,a) - V^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s) - (Q_c^{\pi_{\lambda^*(\theta);\theta}}(s,a) - V_c^{\pi_{\lambda^*(\theta);\theta}}(s))\cdot$$
$$\frac{\int_{s\in\mathcal{S}}\int_{a\in\mathcal{A}} \psi^{\pi_{\lambda^*(\theta);\theta}}(s,a)\left[Q^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s,a) - V^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s)\right] Q_c^{\pi_{\lambda^*(\theta);\theta}}(s,a)dads}{\int_{s\in\mathcal{S}}\int_{a\in\mathcal{A}} \psi^{\pi_{\lambda^*(\theta);\theta}}(s,a)\left[Q^{\pi_{\lambda^*(\theta);\theta}}_{c}(s,a) - V^{\pi_{\lambda^*(\theta);\theta}}_{c}(s)\right] Q_c^{\pi_{\lambda^*(\theta);\theta}}(s,a)dads},$$
$$= Q^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s,a) - V^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s) - C_{\pi_{\lambda^*(\theta);\theta}}(Q_c^{\pi_{\lambda^*(\theta);\theta}}(s,a) - V_c^{\pi_{\lambda^*(\theta);\theta}}(s)), \tag{14}$$

where $C_{\pi_{\lambda;\theta}} \triangleq \frac{\int_{s\in\mathcal{S}}\int_{a\in\mathcal{A}} \psi^{\pi_{\lambda;\theta}}(s,a)\left[Q^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s,a) - V^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s)\right] Q_c^{\pi_{\lambda;\theta}}(s,a)dads}{\int_{s\in\mathcal{S}}\int_{a\in\mathcal{A}} \psi^{\pi_{\lambda;\theta}}(s,a)\left[Q^{\pi_{\lambda;\theta}}_{c}(s,a) - V^{\pi_{\lambda;\theta}}_{c}(s)\right] Q_c^{\pi_{\lambda;\theta}}(s,a)dads}$, (b) follows Lemma 5, and (c) follows (11) and (13). Note that we can equivalently reformulate $C_{\pi_{\lambda;\theta}}$ as:

$$C_{\pi_{\lambda;\theta}} = \frac{E_{(s,a)\sim\psi^{\pi_{\lambda;\theta}}(\cdot,\cdot)}[(Q^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s,a) - V^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s))Q_c^{\pi_{\lambda;\theta}}(s,a)]}{E_{(s,a)\sim\psi^{\pi_{\lambda;\theta}}(\cdot,\cdot)}[(Q^{\pi_{\lambda;\theta}}_{c}(s,a) - V^{\pi_{\lambda;\theta}}_{c}(s))Q_c^{\pi_{\lambda;\theta}}(s,a)]}. \tag{15}$$

Therefore, we can compute the hyper-gradient as:

$$\frac{dJ_r(\pi_{\lambda^*(\theta);\theta})}{d\theta} \overset{(d)}{=} E_{(s,a)\sim\psi^{\pi_{\lambda^*(\theta);\theta}}}\left[\frac{d\log\pi_{\lambda^*(\theta);\theta}}{d\theta} Q_r^{\pi_{\lambda^*(\theta);\theta}}(s,a)\right],$$
$$\overset{(e)}{=} E_{(s,a)\sim\psi^{\pi_{\lambda^*(\theta);\theta}}}\left[\left(Q^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s,a) - V^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s) - C_{\pi_{\lambda^*(\theta);\theta}}(Q_c^{\pi_{\lambda^*(\theta);\theta}}(s,a) - V_c^{\pi_{\lambda^*(\theta);\theta}}(s))\right)\cdot\right.$$
$$\left. Q_r^{\pi_{\lambda^*(\theta);\theta}}(s,a)\right],$$

where (d) follows the standard result of policy gradient (Sutton & Barto, 2018), and (e) follows (14).

D.4 Proof of Lemma 3

Since $\pi_{\tilde{n}+1}(a|s) \propto \exp(Q^{\pi_{\tilde{n}}}_{\lambda;\theta}(s,a))$, from Appendix C, we can see that $\pi_{\tilde{n}+1}(a|s) = \frac{\exp(Q^{\pi_{\tilde{n}}}_{\lambda;\theta}(s,a))}{\exp(V^{\pi_{\tilde{n}}}_{\lambda;\theta}(s))}$.

$$|\log\pi_{\tilde{n}+1}(a|s) - \log\pi_{\lambda;\theta}(a|s)| = |Q^{\pi_{\tilde{n}}}_{\lambda;\theta}(s,a) - V^{\pi_{\tilde{n}}}_{\lambda;\theta}(s) - Q_{\lambda;\theta}(s,a) + V_{\lambda;\theta}(s)|,$$
$$\overset{(a)}{=} Q_{\lambda;\theta}(s,a) - Q^{\pi_{\tilde{n}}}_{\lambda;\theta}(s,a) + V_{\lambda;\theta}(s) - V^{\pi_{\tilde{n}}}_{\lambda;\theta}(s),$$
$$\overset{(b)}{\leq} Q_{\lambda;\theta}(s,a) - \mathcal{T}^{\text{soft}}_{\lambda;\theta}(Q^{\pi_{\tilde{n}-1}}_{\lambda;\theta})(s,a) + V_{\lambda;\theta}(s) - \mathcal{T}^{\text{soft}}_{\lambda;\theta}(V^{\pi_{\tilde{n}-1}}_{\lambda;\theta})(s),$$
$$\overset{(c)}{=} \mathcal{T}^{\text{soft}}_{\lambda;\theta}(Q_{\lambda;\theta})(s,a) - \mathcal{T}^{\text{soft}}_{\lambda;\theta}(Q^{\pi_{\tilde{n}-1}}_{\lambda;\theta})(s,a) + \mathcal{T}^{\text{soft}}_{\lambda;\theta}(V_{\lambda;\theta})(s) - \mathcal{T}^{\text{soft}}_{\lambda;\theta}(V^{\pi_{\tilde{n}-1}}_{\lambda;\theta})(s),$$
$$\overset{(d)}{\leq} \gamma\left[Q_{\lambda;\theta}(s,a) - Q^{\pi_{\tilde{n}-1}}_{\lambda;\theta}(s,a) + V_{\lambda;\theta}(s) - V^{\pi_{\tilde{n}-1}}_{\lambda;\theta}(s)\right],$$
$$\leq \gamma^{\tilde{n}+1}\left[Q_{\lambda;\theta}(s,a) - Q^{\pi_0}_{\lambda;\theta}(s,a) + V_{\lambda;\theta}(s) - V^{\pi_0}_{\lambda;\theta}(s)\right],$$

where (a) follows policy improvement theorem (Theorem 4 in (Haarnoja et al., 2017)) (note that $Q_{\lambda;\theta}$ and $V_{\lambda;\theta}$ are the optimal Q/value functions under (λ, θ)), (b) follows Lemma 7, (c) follows the fact that the optimal Q/value functions are the fixed points of the contraction operator $\mathcal{T}^{\text{soft}}_{\lambda;\theta}$, and (d) follows Lemma 6.

Lemma 8. *For any θ and $\bar{n} \geq 0$, it holds that $\nabla^2_{\lambda\lambda} G(\lambda_{\bar{n}}; \theta) \succeq \tau_G I$ where τ_G is a positive constant.*

Proof. It has been proved in Subsection D.2 that $\nabla^2_{\lambda\lambda} G(\lambda; \theta) \succ 0$. To prove that $\nabla^2_{\lambda\lambda} G(\lambda_{\bar{n}}; \theta) \succeq \tau_G I$, we first prove that $\nabla^2_{\lambda\lambda} G(\lambda; \theta)$ is continuous in λ and then prove that the trajectory of $\lambda_{\bar{n}}$ is bounded within a compact set for any $\bar{n} \geq 0$.

From (11), we know that

$$\nabla^2_{\lambda\lambda} G(\lambda; \theta) = E^{\pi_{\lambda;\theta}}\left[\sum_{t=0}^{\infty} \gamma^t \left(Q_c^{\pi_{\lambda;\theta}}(s,a) - V_c^{\pi_{\lambda;\theta}}(s,a)\right) Q_c^{\pi_{\lambda;\theta}}(s,a)\right].$$

Since $\pi_{\lambda;\theta}$, $Q_c^{\pi_{\lambda;\theta}}(s,a)$, and $V_c^{\pi_{\lambda;\theta}}(s,a)$ are differentiable to λ, we know that $\nabla^2_{\lambda\lambda} G(\lambda; \theta)$ is continuous to λ. Now, we show that the trajectory of $\lambda_{\bar{n}}$ is bounded within a compact set.

$$\begin{aligned}
& ||\lambda_{\bar{n}+1} - \lambda^*(\theta)||^2 = ||\lambda_{\bar{n}} - \alpha_{\bar{n}} g_{\lambda_{\bar{n}};\theta} - \lambda^*(\theta)||^2, \\
& = ||\lambda_{\bar{n}} - \lambda^*(\theta)||^2 + \alpha_{\bar{n}}^2 ||g_{\lambda_{\bar{n}};\theta}||^2 - \alpha_{\bar{n}} \langle g_{\lambda_{\bar{n}};\theta}, \lambda_{\bar{n}} - \lambda^*(\theta) \rangle, \\
& = ||\lambda_{\bar{n}} - \lambda^*(\theta)||^2 + \alpha_{\bar{n}}^2 ||g_{\lambda_{\bar{n}};\theta}||^2 - \alpha_{\bar{n}} \langle \nabla_\lambda G(\lambda_{\bar{n}}; \theta), \lambda_{\bar{n}} - \lambda^*(\theta) \rangle \\
& - \alpha_{\bar{n}} \langle g_{\lambda_{\bar{n}};\theta} - \nabla_\lambda G(\lambda_{\bar{n}}; \theta), \lambda_{\bar{n}} - \lambda^*(\theta) \rangle, \\
& \leq ||\lambda_{\bar{n}} - \lambda^*(\theta)||^2 + \alpha_{\bar{n}}^2 ||g_{\lambda_{\bar{n}};\theta}||^2 - \alpha_{\bar{n}} [G(\lambda_{\bar{n}}; \theta) - G(\lambda^*(\theta); \theta)] \\
& - \alpha_{\bar{n}} \langle g_{\lambda_{\bar{n}};\theta} - \nabla_\lambda G(\lambda_{\bar{n}}; \theta), \lambda_{\bar{n}} - \lambda^*(\theta) \rangle, \qquad (16) \\
& \leq ||\lambda_{\bar{n}} - \lambda^*(\theta)||^2 + \alpha_{\bar{n}}^2 ||g_{\lambda_{\bar{n}};\theta}||^2 - \alpha_{\bar{n}} \langle g_{\lambda_{\bar{n}};\theta} - \nabla_\lambda G(\lambda_{\bar{n}}; \theta), \lambda_{\bar{n}} - \lambda^*(\theta) \rangle, \\
& \leq ||\lambda_{\bar{n}} - \lambda^*(\theta)||^2 + \alpha_{\bar{n}}^2 ||g_{\lambda_{\bar{n}};\theta}||^2 + \alpha_{\bar{n}} ||g_{\lambda_{\bar{n}};\theta} - \nabla_\lambda G(\lambda_{\bar{n}}; \theta)|| \cdot ||\lambda_{\bar{n}} - \lambda^*(\theta)||, \\
& \overset{(a)}{\leq} ||\lambda_{\bar{n}} - \lambda^*(\theta)||^2 + \alpha_{\bar{n}}^2 ||g_{\lambda_{\bar{n}};\theta}||^2 + \alpha_{\bar{n}} C \gamma^{\bar{n}} ||\lambda_{\bar{n}} - \lambda^*(\theta)||, \\
& = ||\lambda_{\bar{n}} - \lambda^*(\theta)||^2 + \alpha_{\bar{n}}^2 ||g_{\lambda_{\bar{n}};\theta}||^2 + \alpha_{\bar{n}} C \gamma^{\bar{n}} ||\lambda_0 - \lambda^*(\theta) - \sum_{i=0}^{\bar{n}-1} \alpha_i g_{\lambda_i;\theta}||, \\
& \leq ||\lambda_{\bar{n}} - \lambda^*(\theta)||^2 + \alpha_{\bar{n}}^2 ||g_{\lambda_{\bar{n}};\theta}||^2 + \alpha_{\bar{n}} C \gamma^{\bar{n}} ||\lambda_0 - \lambda^*(\theta)|| + \alpha_{\bar{n}} C \gamma^{\bar{n}} ||\sum_{i=0}^{\bar{n}-1} \alpha_i g_{\lambda_i;\theta}||, \\
& \overset{(b)}{\leq} ||\lambda_{\bar{n}} - \lambda^*(\theta)||^2 + \alpha_{\bar{n}}^2 (b + \frac{c_{\max}}{1-\gamma})^2 + \alpha_{\bar{n}} C \gamma^{\bar{n}} ||\lambda_0 - \lambda^*(\theta)|| + \alpha_{\bar{n}} C \gamma^{\bar{n}} \sum_{i=0}^{\bar{n}-1} \alpha_i (b + \frac{c_{\max}}{1-\gamma}),
\end{aligned} \tag{17}$$

where (a) follows (18), and (b) follows that $||g_{\lambda;\theta}|| = ||b - J_c(\pi_{\lambda;\theta})|| \leq b + \frac{c_{\max}}{1-\gamma}$. Now we show that $\sum_{\bar{n}=1}^{\infty} \alpha_{\bar{n}} \gamma^{\bar{n}} \sum_{i=0}^{\bar{n}-1} \alpha_i$ is bounded. Since $\alpha_i \propto 1/i^{\bar{\eta}}$, we know that $\sum_{i=0}^{\bar{n}-1} \alpha_i = O(\bar{n}^{1-\bar{\eta}})$. Therefore, we know that $\alpha_{\bar{n}} \sum_{i=0}^{\bar{n}-1} \alpha_i = O(\bar{n}^{1-2\bar{\eta}}) \leq \bar{C}$ where $\bar{C}$ is a positive constant. Therefore, $\sum_{\bar{n}=1}^{\infty} \alpha_{\bar{n}} \gamma^{\bar{n}} \sum_{i=0}^{\bar{n}-1} \alpha_i \leq \bar{C} \sum_{\bar{n}=1}^{\infty} \gamma^{\bar{n}}$ is bounded. Now, we sum the both sides of (17) from $\bar{n} = 1$ to $\bar{N} - 1$:

$$\begin{aligned}
& \sum_{\bar{n}=0}^{\bar{N}-1} ||\lambda_{\bar{n}+1} - \lambda^*(\theta)||^2, \\
& \leq \sum_{\bar{n}=0}^{\bar{N}-1} ||\lambda_{\bar{n}} - \lambda^*(\theta)||^2 + \alpha_{\bar{n}}^2 (b + \frac{c_{\max}}{1-\gamma})^2 + \alpha_{\bar{n}} C \gamma^{\bar{n}} ||\lambda_0 - \lambda^*(\theta)|| + \alpha_{\bar{n}} C \gamma^{\bar{n}} \sum_{i=0}^{\bar{n}-1} \alpha_i (b + \frac{c_{\max}}{1-\gamma}), \\
& \Rightarrow ||\lambda_{\bar{N}} - \lambda^*(\theta)||^2, \\
& \leq ||\lambda_0 - \lambda^*(\theta)||^2 + \sum_{\bar{n}=0}^{\bar{N}-1} \alpha_{\bar{n}}^2 (b + \frac{c_{\max}}{1-\gamma})^2 + \alpha_{\bar{n}} C \gamma^{\bar{n}} ||\lambda_0 - \lambda^*(\theta)|| + \alpha_{\bar{n}} C \gamma^{\bar{n}} \sum_{i=0}^{\bar{n}-1} \alpha_i (b + \frac{c_{\max}}{1-\gamma}),
\end{aligned}$$

$$\leq ||\lambda_0 - \lambda^*(\theta)||^2 + \sum_{\bar{n}=0}^{\infty} \alpha_{\bar{n}}^2 (b + \frac{c_{\max}}{1-\gamma})^2 + \alpha_{\bar{n}} C \gamma^{\bar{n}} ||\lambda_0 - \lambda^*(\theta)|| + \alpha_{\bar{n}} C \gamma^{\bar{n}} \sum_{i=0}^{\bar{n}-1} \alpha_i (b + \frac{c_{\max}}{1-\gamma}).$$

Note that $\alpha_{\bar{n}} \propto \frac{1}{(\bar{n}+1)^{\bar{\eta}}}$ and $\bar{\eta} \in (\frac{1}{2}, 1)$, it is obvious that $|\lambda_{\bar{N}} - \lambda^*(\theta)|^2$ is bounded. Therefore, the trajectory of $\lambda_{\bar{n}}$ is bounded for any $\bar{n} \geq 0$. Therefore, we can always find a positive constant τ_G such that $\nabla^2_{\lambda\lambda} G(\lambda; \theta) \succeq \tau_G I$. ☐

D.5 Proof of Lemma 4

We first quantify the gradient approximation error $|\nabla_\lambda G(\lambda; \theta) - g_{\lambda;\theta}|$ and then show the convergence of the middle loop.

$$\begin{aligned}
|\nabla_\lambda G(\lambda; \theta) - g_{\lambda;\theta}| &= |J_c(\pi_{\lambda;\theta}) - J_c(\hat{\pi}_{\lambda;\theta})|, \\
&= \left| \int_{s\in\mathcal{S}} \int_{a\in\mathcal{A}} \left[\psi^{\pi_{\lambda;\theta}}(s,a) - \psi^{\hat{\pi}_{\lambda;\theta}}(s,a) \right] c(s,a) da ds \right|, \\
&\leq c_{\max} \int_{s\in\mathcal{S}} \int_{a\in\mathcal{A}} \left| \psi^{\pi_{\lambda;\theta}}(s,a) - \psi^{\hat{\pi}_{\lambda;\theta}}(s,a) \right| da ds, \\
&\overset{(a)}{\leq} c_{\max} C_d \int_{s\in\mathcal{S}} \int_{a\in\mathcal{A}} \left| Q_{\lambda;\theta}(s,a) - Q^{\hat{\pi}_{\lambda;\theta}}_{\lambda;\theta}(s,a) \right| da ds, \\
&\leq c_{\max} C_d C_{SA} \max_{(s,a)\in\mathcal{S}\times\mathcal{A}} \{ |Q_{\lambda;\theta}(s,a) - Q^{\hat{\pi}_{\lambda;\theta}}_{\lambda;\theta}(s,a)| \}, \\
&\overset{(b)}{\leq} c_{\max} C_d C_{SA} \gamma^{\tilde{N}_{\bar{n}}} \max_{(s,a)\in\mathcal{S}\times\mathcal{A}} \{ |Q_{\lambda;\theta}(s,a) - Q^{\pi_0}_{\lambda;\theta}(s,a)| \}, \\
&= C \gamma^{\tilde{N}_{\bar{n}}}
\end{aligned} \tag{18}$$

where (a) follows step (iv) of equation (64) in (Zeng et al., 2022) and C_d is a positive constant, C_{SA} can be any positive constant that is larger that the area of $\mathcal{S} \times \mathcal{A}$, (b) follows the proof in Subsection D.4, and $C = c_{\max} C_d C_{SA} \max_{(s,a)\in\mathcal{S}\times\mathcal{A}} \{ |Q_{\lambda;\theta}(s,a) - Q^{\pi_0}_{\lambda;\theta}(s,a)| \}$.

Now, we quantify the convergence of the middle loop. From the expression (11) of $\nabla^2_{\lambda\lambda} G(\lambda; \theta)$, we know that $||\nabla^2_{\lambda\lambda} G(\lambda; \theta)|| \leq \frac{2c^2_{\max}}{(1-\gamma)^2}$. From (16), we know that:

$$\begin{aligned}
&\alpha_{\bar{n}} [G(\lambda_{\bar{n}}; \theta) - G(\lambda^*(\theta); \theta)] \leq ||\lambda_{\bar{n}} - \lambda^*(\theta)||^2 - ||\lambda_{\bar{n}+1} - \lambda^*(\theta)||^2 \\
&+ \alpha_{\bar{n}}^2 ||g_{\lambda_{\bar{n}};\theta}||^2 - \alpha_{\bar{n}} \langle g_{\lambda_{\bar{n}};\theta} - \nabla_\lambda G(\lambda_{\bar{n}}; \theta), \lambda_{\bar{n}} - \lambda^*(\theta) \rangle, \\
&\leq ||\lambda_{\bar{n}} - \lambda^*(\theta)||^2 - ||\lambda_{\bar{n}+1} - \lambda^*(\theta)||^2 + \alpha_{\bar{n}}^2 ||g_{\lambda_{\bar{n}};\theta}||^2 + \alpha_{\bar{n}} ||g_{\lambda_{\bar{n}};\theta} - \nabla_\lambda G(\lambda_{\bar{n}}; \theta)|| \cdot ||\lambda_{\bar{n}} - \lambda^*(\theta)||, \\
&\overset{(c)}{\leq} ||\lambda_{\bar{n}} - \lambda^*(\theta)||^2 - ||\lambda_{\bar{n}+1} - \lambda^*(\theta)||^2 + \alpha_{\bar{n}}^2 ||g_{\lambda_{\bar{n}};\theta}||^2 + \alpha_{\bar{n}} \gamma^{\bar{n}} \tilde{C}, \\
&\overset{(d)}{\leq} ||\lambda_{\bar{n}} - \lambda^*(\theta)||^2 - ||\lambda_{\bar{n}+1} - \lambda^*(\theta)||^2 + \alpha_{\bar{n}}^2 (b + \frac{c_{\max}}{1-\gamma})^2 + \alpha_{\bar{n}} \gamma^{\bar{n}} \tilde{C},
\end{aligned} \tag{19}$$

where (c) follows (18) and the fact that $||\lambda_{\bar{n}} - \lambda^*(\theta)||$ is bounded (proved in Lemma 8), $\tilde{C}$ is a positive constant, and (d) follows that $||g_{\lambda;\theta}|| = ||b - J_c(\pi_{\lambda;\theta})|| \leq b + \frac{c_{\max}}{1-\gamma}$. Telescoping (19) from $\bar{n} = 0$ to $\bar{N} - 1$:

$$\begin{aligned}
&\sum_{\bar{n}=0}^{\bar{N}-1} \alpha_{\bar{n}} [G(\lambda_{\bar{n}}; \theta) - G(\lambda^*(\theta); \theta)], \\
&\leq ||\lambda_0 - \lambda^*(\theta)||^2 - ||\lambda_{\bar{N}} - \lambda^*(\theta)||^2 + \sum_{\bar{n}=0}^{\bar{N}-1} \alpha_{\bar{n}}^2 (b + \frac{c_{\max}}{1-\gamma})^2 + \sum_{\bar{n}=0}^{\bar{N}-1} \alpha_{\bar{n}} \gamma^{\bar{n}} \tilde{C}.
\end{aligned}$$

Since $\alpha_{\bar{n}} = \frac{1}{(\bar{n}+1)^{\bar{\eta}}}$ and $\bar{\eta} \in (\frac{1}{2}, 1)$, there is a positive constant $D_{\max}$ such that $\sum_{\bar{n}=0}^{\bar{N}} \alpha_{\bar{n}}^2 (b + \frac{c_{\max}}{1-\gamma})^2 + \sum_{\bar{n}=0}^{\bar{N}} \alpha_{\bar{n}} \gamma^{\bar{n}-1} \tilde{C} \leq D_{\max}$. Therefore, we have that

$$\sum_{\bar{n}=0}^{\bar{N}-1} \frac{1}{\bar{N}^{\bar{\eta}}} [G(\lambda_{\bar{n}}; \theta) - G(\lambda^*(\theta); \theta)] \leq \sum_{\bar{n}=0}^{\bar{N}-1} \alpha_{\bar{n}} [G(\lambda_{\bar{n}}; \theta) - G(\lambda^*(\theta); \theta)],$$

$$
\begin{aligned}
&\leq ||\lambda_0 - \lambda^*(\theta)||^2 - ||\lambda_{\bar{N}} - \lambda^*(\theta)||^2 + D_{\max}, \\
&\Rightarrow \frac{1}{\bar{N}} \sum_{\bar{n}=0}^{\bar{N}-1} [G(\lambda_{\bar{n}};\theta) - G(\lambda^*(\theta);\theta)] \leq \frac{1}{N^{1-\bar{\eta}}} \Big[||\lambda_0 - \lambda^*(\theta)||^2 - ||\lambda_{\bar{N}} - \lambda^*(\theta)||^2 + D_{\max} \Big].
\end{aligned} \tag{20}
$$

Therefore, we have that

$$
\begin{aligned}
&||\hat{\lambda}(\theta) - \lambda^*(\theta)|| \overset{(e)}{\leq} \frac{2}{\tau_G}[G(\hat{\lambda}(\theta);\theta) - G(\lambda^*(\theta);\theta)] \overset{(f)}{\leq} \frac{2}{\tau_G}[\frac{1}{\bar{N}} \sum_{\bar{n}=0}^{\bar{N}-1} G(\lambda_{\bar{n}};\theta) - G(\lambda^*(\theta);\theta)], \\
&\overset{(g)}{\leq} O(\frac{1}{\bar{N}^{1-\bar{\eta}}}),
\end{aligned} \tag{21}
$$

where (e) follows the fact that $G(\lambda;\theta)$ is τ_G-strongly convex (Lemma 8), (f) follows Jensen's inequality (note that $\hat{\lambda}(\theta) = \frac{1}{\bar{N}}\sum_{\bar{n}=0}^{\bar{N}-1}\lambda_{\bar{n}}$), and (g) follows (20).

Now, we take a look at the term

$$
\begin{aligned}
&|\log \pi_{\lambda^*(\theta);\theta}(a|s) - \log \hat{\pi}_{\hat{\lambda}(\theta);\theta}(a|s)|, \\
&\leq |\log \pi_{\lambda^*(\theta);\theta}(a|s) - \log \pi_{\hat{\lambda}(\theta);\theta}(a|s)| + |\log \pi_{\hat{\lambda}(\theta);\theta}(a|s)| - \log \hat{\pi}_{\hat{\lambda}(\theta);\theta}(a|s)|, \\
&\overset{(h)}{\leq} \frac{2c_{\max}}{1-\gamma}||\hat{\lambda}(\theta) - \lambda^*(\theta)|| + |\log \pi_{\hat{\lambda}(\theta);\theta}(a|s)| - \log \hat{\pi}_{\hat{\lambda}(\theta);\theta}(a|s)|, \\
&\overset{(i)}{\leq} O(\frac{1}{\bar{N}^{1-\bar{\eta}}} + \gamma^{\bar{N}}),
\end{aligned} \tag{22}
$$

where (h) follows Lemma 5 such that $|\nabla_\lambda \log \pi_{\lambda;\theta}| \leq \frac{2c_{\max}}{1-\gamma}$, and (i) follows (21) and Lemma 3.

Lemma 9. *The upper-level loss function $J_r(\pi_{\lambda^*(\theta);\theta})$ is L-Lipschitz and $\bar{L}$-smooth where L and $\bar{L}$ are positive constants. Moreover, it holds that $||g_\theta|| \leq L$ and $||\nabla_\theta g_\theta|| \leq \bar{L}$.*

Proof. This suffices to show that the norms $||\nabla J_r(\pi_{\lambda^*(\theta);\theta})||$ and $||\nabla^2 J_r(\pi_{\lambda^*(\theta);\theta})||$ are upper bounded by L and $\bar{L}$. From Subsection D.3, we know that

$$
\begin{aligned}
&\frac{dJ_r(\pi_{\lambda^*(\theta);\theta})}{d\theta}, \\
&= E_{(s,a)\sim\psi^{\pi_{\lambda^*(\theta);\theta}}}\Bigg[\Bigg(Q^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s,a) - V^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s) - C_{\pi_{\lambda^*(\theta);\theta}}(Q^{\pi_{\lambda^*(\theta);\theta}}_c(s,a) - V^{\pi_{\lambda^*(\theta);\theta}}_c(s))\Bigg)\cdot \\
&Q^{\pi_{\lambda^*(\theta);\theta}}_r(s,a)\Bigg],
\end{aligned}
$$

where $C_{\pi^*_\lambda(\theta);\theta} = [\nabla^2_{\lambda\lambda}G(\lambda^*(\theta);\theta)]^{-1}\nabla^2_{\lambda\theta}G(\lambda^*(\theta);\theta)$. Since $||[\nabla^2_{\lambda\lambda}G(\lambda^*(\theta);\theta)]^{-1}|| \leq \frac{1}{\tau_G}$ (Lemma 8) and $||\nabla^2_{\lambda\theta}G(\lambda;\theta)|| = ||E_{(s,a)\sim\psi^{\pi_{\lambda;\theta}}(\cdot,\cdot)}[(Q^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s,a) - V^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s))Q^{\pi_{\lambda;\theta}}_c(s,a)]|| \leq \frac{2C_2c_{\max}}{(1-\gamma)^2}$, we know that $||C_{\pi_{\lambda^*(\theta);\theta}}|| \leq \frac{2C_2c_{\max}}{(1-\gamma)^2\tau_G}$. Therefore, it holds that

$$
||\frac{dJ_r(\pi_{\lambda^*(\theta);\theta})}{d\theta}|| \leq \frac{1}{1-\gamma}\cdot\Bigg[\Bigg(\frac{2C_2}{1-\gamma} + \frac{2C_2c_{\max}}{(1-\gamma)^2\tau_G}\cdot\frac{2c_{\max}}{1-\gamma}\Bigg)\frac{C_1}{1-\gamma}\Bigg] = L. \tag{23}
$$

Similarly, we can see that $||g_\theta|| \leq L$.

Now, we take a look at the Hessian term $\nabla^2 J_r(\pi_{\lambda^*(\theta);\theta})$. We define $h^\pi(s,a) \triangleq Q^\pi_{\nabla_\theta r_\theta}(s_t,a_t) - V^\pi_{\nabla_\theta r_\theta}(s_t) - C_\pi(Q^\pi_c(s_t,a_t) - V^\pi_c(s_t))]Q^\pi_r(s_t,a_t)$, $H^\pi(s,a) \triangleq E^\pi[\sum_{t=0}^\infty \gamma^t h^\pi(s_t,a_t)|s_0 = s, a_0 = a]$, and $H^\pi(s) \triangleq E^\pi[\sum_{t=0}^\infty \gamma^t h^\pi(s_t,a_t)|s_0 = s]$. We know that $||H^\pi(s,a)|| \leq L$ and $||H^\pi(s)|| \leq L$. Therefore, we have that

$$
\nabla^2 J_r(\pi_{\lambda^*(\theta);\theta}) = \nabla\int_{s_0\in\mathcal{S}} P_0(s_0)\int_{a_0\in\mathcal{A}} \pi_{\lambda^*(\theta);\theta}(a_0|s_0)H^{\pi_{\lambda^*(\theta);\theta}}(s_0,a_0)da_0ds_0,
$$

$$= \int_{s_0 \in \mathcal{S}} P_0(s_0) \int_{a_0 \in \mathcal{A}} \Big[\nabla \pi_{\lambda^*(\theta);\theta}(a_0|s_0) \cdot H^{\pi_{\lambda^*(\theta);\theta}}(s_0, a_0)$$

$$+ \pi_{\lambda^*(\theta);\theta}(a_0|s_0) \cdot \nabla H^{\pi_{\lambda^*(\theta);\theta}}(s_0, a_0) da_0 ds_0 \Big],$$

$$= \int_{s_0 \in \mathcal{S}} P_0(s_0) \int_{a_0 \in \mathcal{A}} \Big[\pi_{\lambda^*(\theta);\theta}(a_0|s_0) \nabla \log \pi_{\lambda^*(\theta);\theta}(a_0|s_0) \cdot H^{\pi_{\lambda^*(\theta);\theta}}(s_0, a_0)$$

$$+ \pi_{\lambda^*(\theta);\theta}(a_0|s_0) \cdot \Big(\nabla h^{\pi_{\lambda^*(\theta);\theta}}(s_0, a_0) + \gamma \int_{s_1 \in \mathcal{S}} P(s_1|s_0, a_0) \nabla H^{\pi_{\lambda^*(\theta);\theta}}(s_1) ds_1 \Big) da_0 ds_0 \Big].$$

Keep the expansion, we know that

$$\nabla^2 J_r(\pi_{\lambda^*(\theta);\theta}),$$
$$= \int_{(s,a) \in \mathcal{S} \times \mathcal{A}} \psi^{\pi_{\lambda^*(\theta);\theta}}(s, a) \Big[\nabla h^{\pi_{\lambda^*(\theta);\theta}}(s, a) + \nabla \log \pi_{\lambda^*(\theta);\theta}(a|s) \cdot H^{\pi_{\lambda^*(\theta);\theta}}(s, a) \Big] dads, \tag{24}$$

Since $\nabla \log \pi_{\lambda^*(\theta);\theta}(a|s)$ and $H^{\pi_{\lambda^*(\theta);\theta}}(s, a)$ are both bounded, the only thing left is to bound $||\nabla h^{\pi_{\lambda^*(\theta);\theta}}(s, a)||$. We aim to bound $||\nabla h^{\pi_{\lambda^*(\theta);\theta}}(s, a)||$ by bounding each term in $||h^{\pi_{\lambda^*(\theta);\theta}}(s, a)||$. Note that $\nabla C_{\pi_{\lambda^*(\theta);\theta}} = \nabla([\nabla^2_{\lambda\lambda} G(\lambda^*(\theta); \theta)]^{-1} \nabla^2_{\lambda\theta} G(\lambda^*(\theta); \theta)) = [\nabla^2_{\lambda\lambda} G(\lambda^*(\theta); \theta)]^{-2} \nabla^2_{\lambda\theta} G(\lambda^*(\theta); \theta) \frac{d}{d\theta}(\nabla^2_{\lambda\lambda} G(\lambda^*(\theta); \theta)) + [\nabla^2_{\lambda\lambda} G(\lambda^*(\theta); \theta)]^{-1} \frac{d}{d\theta} \nabla^2_{\lambda\theta} G(\lambda^*(\theta); \theta)$. Therefore, it suffices to show that $||\frac{d}{d\theta} Q^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s, a)||$ and $||\frac{d}{d\theta} Q^{\pi_{\lambda^*(\theta);\theta}}_c(s, a)||$ are bounded.

$$\frac{d}{d\theta} Q^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s, a) = \frac{d}{d\theta} E^{\pi^*_\lambda(\theta);\theta}[\sum_{t=0}^{\infty} \gamma^t \nabla_\theta r_\theta(s_t|a_t)|s_0 = s, a_0 = a],$$

$$= \frac{d}{d\theta} \int_{s_0 \in \mathcal{S}} P_0(s_0) \int_{s_0 \in \mathcal{A}} \Big[\nabla \pi_{\lambda^*(\theta);\theta}(a_0|s_0) \cdot Q^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s_0, a_0)$$

$$+ \pi_{\lambda^*(\theta);\theta}(a_0|s_0) \cdot \nabla Q^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s_0, a_0) \Big] da_0 ds_0,$$

$$= \frac{d}{d\theta} \int_{s_0 \in \mathcal{S}} P_0(s_0) \int_{s_0 \in \mathcal{A}} \Big[\pi_{\lambda^*(\theta);\theta}(a_0|s_0) \nabla \log \pi_{\lambda^*(\theta);\theta}(a_0|s_0) \cdot Q^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s_0, a_0)$$

$$+ \pi_{\lambda^*(\theta);\theta}(a_0|s_0) \cdot \Big(\nabla^2_{\theta\theta} r_\theta(s_0, a_0) + \gamma \int_{s_1 \in \mathcal{S}} P(s_1|s_0, a_0) \nabla V^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s_1) ds_1 \Big) \Big] da_0 ds_0.$$

Keep the expansion, we can get that

$$\frac{d}{d\theta} Q^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s, a),$$
$$= \int_{(s,a) \in \mathcal{S} \times \mathcal{A}} \psi^{\pi_{\lambda^*(\theta);\theta}} \Big[\nabla^2_{\theta\theta} r_\theta(s_0, a_0) + \nabla \log \pi_{\lambda^*(\theta);\theta}(a_0|s_0) \cdot Q^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s_0, a_0) \Big] dads.$$

Therefore, we can see that $||\frac{d}{d\theta} Q^{\pi_{\lambda^*(\theta);\theta}}_{\nabla_\theta r_\theta}(s, a)|| \leq \frac{C_3}{1-\gamma} + \Big[\frac{2c_{\max}}{(1-\gamma)} \cdot \frac{C_2}{\tau_G(1-\gamma)} + \frac{2C_2}{(1-\gamma)} \Big] \cdot \frac{C_2}{(1-\gamma)^2}$. Similarly, we can also see that $\frac{d}{d\theta} Q^{\pi_{\lambda^*(\theta);\theta}}_c(s, a)$ is also bounded. Therefore, we can find a positive constant $\bar{L}$ such that $||\nabla^2 J_r(\pi_{\lambda^*(\theta);\theta})|| \leq \bar{L}$. With the same procedure, we can see that $||\nabla_\theta g_\theta|| \leq \bar{L}$. □

Lemma 10. *The hyper-gradient approximation error is upper bounded, i.e.,* $||\frac{d}{d\theta} J_r(\pi_{\lambda^*(\theta);\theta}) - g_\theta|| \leq O(\gamma^{\bar{N}} + \frac{1}{N^{1-\bar{\eta}}})$.

Proof.

$$||\frac{d}{d\theta} J_r(\pi_{\lambda^*(\theta);\theta}) - g_\theta||,$$
$$\leq \Big|\Big| \int_{(s,a) \in \mathcal{S} \times \mathcal{A}} \Big[\psi^{\hat{\pi}_{\hat{\lambda}(\theta);\theta}}(s, a) h^{\hat{\pi}_{\hat{\lambda}(\theta);\theta}}(s, a) - \psi^{\pi_{\lambda^*(\theta);\theta}}(s, a) h^{\pi_{\lambda^*(\theta);\theta}}(s, a) \Big] dads \Big|\Big|,$$

$$
\begin{aligned}
&\overset{(a)}{\le} (1-\gamma)L\int_{(s,a)\in\mathcal{S}\times\mathcal{A}} ||\psi^{\hat{\pi}_{\hat{\lambda}(\theta);\theta}}(s,a)-\psi^{\pi_{\lambda^*(\theta);\theta}}(s,a)||dads,\\
&\le (1-\gamma)L\int_{(s,a)\in\mathcal{S}\times\mathcal{A}} ||\psi^{\hat{\pi}_{\hat{\lambda}(\theta);\theta}}(s,a)-\psi^{\pi_{\hat{\lambda}(\theta);\theta}}(s,a)||+||\psi^{\pi_{\hat{\lambda}(\theta);\theta}}(s,a)-\psi^{\pi_{\lambda^*(\theta);\theta}}(s,a)||dads,\\
&\overset{(b)}{\le} (1-\gamma)C_dC_{SA}\Big[\max\{|Q_{\hat{\lambda}(\theta);\theta}(s,a)-Q^{\hat{\pi}_{\hat{\lambda}(\theta);\theta}}_{\hat{\lambda}(\theta);\theta}(s,a)|\}+\max\{|Q_{\hat{\lambda}(\theta);\theta}(s,a)-Q_{\lambda^*(\theta);\theta}(s,a)|\}\Big],\\
&\overset{(c)}{\le} (1-\gamma)C_dC_{SA}\Big[\gamma^{\bar{N}}\max\{|Q_{\hat{\lambda}(\theta);\theta}(s,a)-Q^{\pi_0}_{\hat{\lambda}(\theta);\theta}(s,a)|\}\\
&+\max\{|Q_{\hat{\lambda}(\theta);\theta}(s,a)-Q_{\lambda^*(\theta);\theta}(s,a)|\}\Big],\\
&\overset{(d)}{\le} (1-\gamma)C_dC_{SA}\Big[\gamma^{\bar{N}}\max\{|Q_{\hat{\lambda}(\theta);\theta}(s,a)-Q^{\pi_0}_{\hat{\lambda}(\theta);\theta}(s,a)|\}+\frac{c_{\max}}{1-\gamma}||\lambda^*(\theta)-\hat{\lambda}(\theta)||\Big],\\
&\overset{(e)}{\le} O(\gamma^{\bar{N}}+\frac{1}{\bar{N}^{1-\bar{\eta}}}),
\end{aligned}
$$

where (a) follows (23), (b) follows step (iv) of equation (64) in (Zeng et al., 2022), (c) follows (18), (d) follows the fact that $||\nabla_\lambda Q_{\lambda;\theta}(s,a)||=|Q_c^{\pi_{\lambda;\theta}}(s,a)||\le\frac{c_{\max}}{1-\gamma}$, and (e) follows Lemma 4. □

D.6 Proof of Theorem 3

We define a function $f(\theta)$ such that $\nabla f(\theta)=g_\theta$ and $\nabla^2 f(\theta)=\nabla_\theta g_\theta$, therefore, we have that

$$
\begin{aligned}
f(\theta_{n+1}) &\le f(\theta_n)+\langle\nabla f(\theta_n),\theta_{n+1}-\theta_n\rangle+\frac{\bar{L}}{2}||\theta_{n+1}-\theta_n||^2,\\
&= f(\theta_n)-\beta_n||\nabla f(\theta_n)||^2+\frac{\bar{L}\beta_n^2}{2}||\nabla f(\theta_n)||^2,\\
&\Rightarrow \beta_n||\nabla f(\theta_n)||^2\le f(\theta_n)-f(\theta_{n+1})+\frac{\bar{L}\beta_n^2}{2}||\nabla f(\theta_n)||^2. \qquad (25)
\end{aligned}
$$

Telescoping (25) from $n=0$ to $N-1$, we have that

$$
\begin{aligned}
&\sum_{n=0}^{N-1}\beta_n||\nabla f(\theta_n)||^2\le f(\theta_0)-f(\theta_N)+\sum_{n=0}^{N-1}\frac{\bar{L}\beta_n^2}{2}||\nabla f(\theta_n)||^2,\\
&\Rightarrow \sum_{n=0}^{N-1}\frac{1}{N^\eta}||\nabla f(\theta_n)||^2\le\sum_{n=0}^{N-1}\beta_n||\nabla f(\theta_n)||^2\overset{(a)}{\le} f(\theta_0)-f(\theta_N)+\sum_{n=0}^{\infty}\frac{L^2\bar{L}\beta_n^2}{2},\\
&\Rightarrow \frac{1}{N}\sum_{n=0}^{N-1}||\nabla f(\theta_n)||^2\le\frac{1}{N^{1-\eta}}\Big(f(\theta_0)-f(\theta_N)+\sum_{n=0}^{\infty}\frac{L^2\bar{L}\beta_n^2}{2}\Big)\overset{(b)}{=}O(\frac{1}{N^{1-\eta}}), \qquad (26)
\end{aligned}
$$

where (a) follows Lemma 9, and (b) follows the fact that $\sum_{n=0}^{\infty}\beta_n^2$ is bounded as $\beta_n=\frac{1}{(n+1)^\eta}$ and $\eta\in(\frac{1}{2},1)$. Therefore, we have that

$$
\begin{aligned}
\frac{1}{N}\sum_{n=0}^{N-1}||\nabla J_r(\pi_{\lambda^*(\theta);\theta})||^2 &\le \frac{1}{N}\sum_{n=0}^{N-1}\Big(||\nabla f(\theta_n)||^2+||\nabla f(\theta_n)-\nabla J_r(\pi_{\lambda^*(\theta);\theta})||^2\Big),\\
&\overset{(c)}{\le} O(\frac{1}{N^{1-\eta}}+\gamma^{2\bar{N}}+\frac{1}{\bar{N}^{2-2\bar{\eta}}}),
\end{aligned}
$$

where (c) follows (26) and Lemma 10.

D.7 Proof of Theorem 4

We first define $F(\theta)\triangleq J_r(\pi_{\lambda^*(\theta);\theta})$. Theorem 10 in (Agarwal et al., 2021) shows that the policy gradient method can achieve global optimality asymptotically under softmax policy parameterization. The constrained soft policy $\pi_{\lambda^*(\theta);\theta}=\lim_{\bar{N}\to\infty}\hat{\pi}_{\hat{\lambda}(\theta);\theta}$ can be regarded as a softmax policy parameterized by $Q_{\lambda^*(\theta);\theta}$ but the decision variable in our case is θ instead of $Q_{\lambda^*(\theta);\theta}$. However, we can still build on Theorem 10 in (Agarwal et al., 2021) by building connections between θ and $Q_{\lambda^*(\theta);\theta}$. In specific, in order to use the result of Theorem 10 in (Agarwal et al., 2021), we need to prove (i) $F(\theta_n)$ is monotonically increasing, i.e., $F(\theta_{n+1}) \geq F(\theta_n)$ for any $n \geq 0$; (ii) $\frac{dF(\bar{\theta})}{dQ_{\lambda^*(\bar{\theta});\bar{\theta}}} = 0$ if $\frac{dF(\bar{\theta})}{d\theta} = 0$ where $Q_{\lambda^*(\theta);\theta}$ is a vector with the length $|\mathcal{S}| \times |\mathcal{A}|$ whose components are $\{Q_{\lambda^*(\theta);\theta}(s,a)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$. Once proving these two, given that $\lim_{N\to\infty}\lim_{\bar{N}\to\infty}\frac{dF(\theta_N)}{d\theta} = 0$ from Theorem 3, we can use Theorem 10 in (Agarwal et al., 2021) to prove Theorem 4.

Now, we first show that $F(\theta_n)$ is monotonically increasing. This is a straightforward result of Theorem 10.15 in (Beck, 2017) if we choose $\beta_n \leq \frac{1}{L}$. Note that Theorem 10 in (Agarwal et al., 2021) requires $\beta_n \leq \frac{(1-\gamma)^3}{8}$, so that we can choose $\beta_n \leq \min\{\frac{1}{L}, \frac{(1-\gamma)^3}{8}\}$. We next show that $\frac{dF(\bar{\theta})}{dQ_{\lambda^*(\bar{\theta});\bar{\theta}}} = 0$ if $\frac{dF(\bar{\theta})}{d\theta} = 0$.

We know that $\frac{dF(\bar{\theta})}{d\theta} = \frac{dF(\bar{\theta})}{dQ_{\lambda^*(\bar{\theta});\bar{\theta}}} \cdot \frac{dQ_{\lambda^*(\bar{\theta});\bar{\theta}}}{d\theta}$, so that it suffices to show that $\frac{dQ_{\lambda^*(\bar{\theta});\bar{\theta}}(s,a)}{d\theta} \neq 0$ for any (s,a) and any θ. Therefore, we have that

$$\begin{aligned}\frac{dQ_{\lambda^*(\bar{\theta});\bar{\theta}}(s,a)}{d\theta} &= \nabla_\lambda Q_{\lambda^*(\bar{\theta});\bar{\theta}}(s,a)\nabla\lambda^*(\bar{\theta}) + \nabla_\theta Q_{\lambda^*(\bar{\theta});\bar{\theta}}(s,a),\\ &\overset{(a)}{=} Q^{\pi_{\lambda^*(\bar{\theta});\bar{\theta}}}_{\nabla_\theta r_\theta}(s,a) - Q_c^{\pi_{\lambda^*(\bar{\theta});\bar{\theta}}}(s,a)\nabla\lambda^*(\bar{\theta}),\end{aligned} \tag{27}$$

where (a) follows (6) and (8). Now, we first prove that the term $Q^{\pi_{\lambda^*(\bar{\theta});\bar{\theta}}}_{\nabla_\theta r_\theta}(s,a) = E^{\pi_{\lambda^*(\theta);\theta}}[\sum_{t=0}^\infty \gamma^t \nabla_\theta r_{\bar{\theta}}(s_t,a_t)]$ is nonzero. Define $l(\theta;\lambda,s,a) \triangleq E^{\pi_{\lambda;\theta}}[\sum_{t=0}^\infty \gamma^t r_\theta(s_t,a_t)|s_0 = s, a_0 = a]$, and therefore $\nabla_\theta l(\theta;\lambda,s,a) \triangleq E^{\pi_{\lambda;\theta}}[\sum_{t=0}^\infty \gamma^t r_\theta(s_t,a_t)|s_0 = s, a_0 = a]$. We use $\psi^\pi(s'|s,a)$ and $\psi^\pi(s',a'|s,a)$ to denote the state and state-action visitation frequency when the initial state-action is (s,a). Now, we take a look at the Hessian term $\nabla^2_{\theta\theta}l(\theta;\lambda,s,a)$:

$$\begin{aligned}&\nabla^2_{\theta\theta}l(\theta;\lambda,s,a),\\ &= \nabla^2_{\theta\theta}r_\theta(s,a) + \gamma\nabla_\theta\int_{s_1\in\mathcal{S}} P(s_1|s,a)\int_{a_1\in\mathcal{A}}\pi_{\lambda;\theta}(a_1|s_1)\nabla_\theta l(\theta;\lambda,s_1,a_1)da_1 ds_1,\\ &= \nabla^2_{\theta\theta}r_\theta(s,a) + \gamma\int_{s_1\in\mathcal{S}} P(s_1|s,a)\int_{a_1\in\mathcal{A}}\Big[\nabla_\theta\pi_{\lambda;\theta}(a_1|s_1)\cdot\nabla_\theta l(\theta;\lambda,s_1,a_1)\\ &+ \pi_{\lambda;\theta}(a_1|s_1)\cdot\nabla^2_{\theta\theta}l(\theta;\lambda,s_1,a_1)\Big]da_1 ds_1.\end{aligned}$$

Keep the expansion and note that $\nabla_\theta l(\theta;\lambda,s,a) = Q^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s,a)$, we can get that

$$\begin{aligned}\nabla^2_{\theta\theta}l(\theta;\lambda,s,a) &= \int_{s'\in\mathcal{S}}\int_{a'\in\mathcal{A}}\psi^{\pi_{\lambda;\theta}}(s'|s,a)\nabla_\theta\pi_{\lambda;\theta}(a'|s')\cdot Q^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s',a')da'ds',\\ &= \int_{s'\in\mathcal{S}}\int_{a'\in\mathcal{A}}\psi^{\pi_{\lambda;\theta}}(s'|s,a)\pi_{\lambda;\theta}(a'|s')\nabla_\theta\log\pi_{\lambda;\theta}(a'|s')\cdot Q^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s',a')da'ds',\\ &\overset{(b)}{=}\int_{s'\in\mathcal{S}}\psi^{\pi_{\lambda;\theta}}(s'|s,a)\int_{a'\in\mathcal{A}}\pi_{\lambda;\theta}(a'|s')\Big[Q^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s',a') - V^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s')\Big]Q^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s',a')da'ds',\end{aligned} \tag{28}$$

where (b) follows Lemma 5. Note that $V^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s') = E_{a'\sim\pi_{\lambda;\theta}(\cdot|s')}[Q^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s',a')]$. If we use the random variable $Y_{s'a'}$ to denote $Q^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s',a')$, then its expectation is $E_{a'\sim\pi_{\lambda;\theta}(\cdot|s')}(Y_{s'a'}) = V^{\pi_{\lambda;\theta}}_{\nabla_\theta r_\theta}(s')$. We know that the variance $Var(Y_{s'a'}) = E[Y_{s'a'}[Y_{s'a'} - E(Y_{s'a'})]]$. Therefore, we can see that the equation (28) is actually a variance:

$$\nabla^2_{\theta\theta}l(\theta;\lambda,s,a) = \int_{s'\in\mathcal{S}}\psi^{\pi_{\lambda;\theta}}(s'|s,a)Var(Y_{s'a'})ds' \succeq 0.$$

Therefore, the function $l(\theta;\lambda,s,a)$ is convex in θ for any (λ,s,a). If $\nabla_\theta l(\bar{\theta};\lambda,s,a) = 0$, this means that $l(\bar{\theta};\lambda,s,a)$ achieves its optimum. However, $l(\theta;\lambda,s,a)$ does not have an optimum, i.e., $l(\theta;\lambda,s,a)$ can be infinitely large or infinitely small because $r_\theta(s,a)$ can be any arbitrarily large value. This is a contradiction, therefore, $l(\bar{\theta}; \lambda, s, a) \neq 0$ for any (λ, s, a). Then, $l(\bar{\theta}; \lambda^*(\bar{\theta}), s, a) = Q^{\pi_{\lambda^*(\bar{\theta});\bar{\theta}}}_{\nabla_\theta r_\theta}(s, a) \neq 0$.

Recall from (27) that $\frac{dQ_{\lambda^*(\bar{\theta});\bar{\theta}}(s,a)}{d\theta} = Q^{\pi_{\lambda^*(\bar{\theta});\bar{\theta}}}_{\nabla_\theta r_\theta}(s, a) - Q^{\pi_{\lambda^*(\bar{\theta});\bar{\theta}}}_{c}(s, a)\nabla\lambda^*(\bar{\theta})$. For any $(s, a) \notin \mathcal{C}$, $Q^{\pi_{\lambda^*(\bar{\theta});\bar{\theta}}}_{c}(s, a) = 0$ because the policy $\pi_{\lambda^*(\bar{\theta});\bar{\theta}}$ satisfies the constraint of the lower-level problem in (1), i.e., avoiding the set $\mathcal{C}$. Therefore, $\frac{dQ_{\lambda^*(\bar{\theta});\bar{\theta}}(s,a)}{d\theta} = Q^{\pi_{\lambda^*(\bar{\theta});\bar{\theta}}}_{\nabla_\theta r_\theta}(s, a) \neq 0$. For any $(s, a) \in \mathcal{C}$, we know that $Q^{\pi_{\lambda^*(\bar{\theta});\bar{\theta}}}_{c}(s, a) = c(s, a)$ because the policy $\pi_{\lambda^*(\bar{\theta});\bar{\theta}}$ avoids the set $\mathcal{C}$ unless its starting state-action pair is in $\mathcal{C}$. Therefore, we can also design $c(s, a)$ such that $Q^{\pi_{\lambda^*(\bar{\theta});\bar{\theta}}}_{\nabla_\theta r_\theta}(s, a) - c(s, a)\nabla\lambda^*(\bar{\theta}) \neq 0$ for $(s, a) \in \mathcal{C}$. Therefore, we can ensure that $\frac{dQ_{\lambda^*(\bar{\theta});\bar{\theta}}(s,a)}{d\theta} \neq 0$ for any $(s, a) \in \mathcal{S} \times \mathcal{A}$. Therefore, $\frac{dF(\bar{\theta})}{dQ_{\lambda^*(\bar{\theta});\bar{\theta}}} = 0$ if $\frac{dF(\bar{\theta})}{d\theta} = 0$.

E Related works

XRL methods that has the potential to be used to improve RL performance. There are some XRL methods that have the potential to improve the RL performance even if they do not mention that they can improve the RL performance. Value-max (Amir & Amir, 2018; Huang et al., 2018) use the value function $V(s)$ to identify the states with highest value as critical points. We can perturb the actions on these critical states to improve the RL performance.

Constrained reinforcement learning (CRL). The lower-level problem in (1) is a CRL problem. The current works on CRL have two major categories: primal-dual approach and primal approach. The primal-dual approach (Achiam et al., 2017; Tessler et al., 2018; Stooke et al., 2020) converts the CRL problem into an unconstrained optimization problem by using the dual method. Our approach can be categorized as a primal-dual approach. The primal approach (Liu et al., 2020; Chow et al., 2018; Xu et al., 2021) enforce constraints via various designs of the objective function or the update process without an introduction of dual variables. However, these previous methods on CRL may not be suitable to the context of constrained bi-level optimization because they cannot guarantee that the upper-level problem in (1) is smooth after the lower-level problem is solved. The non-smoothness of the upper-level problem can make the constrained bi-level optimization problem difficult to solve. In contrast, our approach ensures the smoothness of the upper-level problem in (1) because we derive an analytical solution of the constrained soft policy and this policy is smooth w.r.t. θ.

E.1 Comparison with (Hu et al., 2020)

Paper (Hu et al., 2020) studies how to utilize the domain knowledge to learn a shaping reward function and formulate a bi-level optimization problem. Here, we discussion our distinctions and improvements compared to (Hu et al., 2020) in terms of problem, assumption, algorithm, and theoretical analysis.

Problem: Our problem is not only to shape the reward function, but also discourage from visiting the set $\mathcal{C}$, therefore, we have a constrained bi-level optimization problem where the lower level is a constrained RL that is non-convex. Paper (Hu et al., 2020) only studies how to shape the reward so that its bi-level optimization problem is unconstrained, which is much easier to solve.

Assumption: Paper (Hu et al., 2020) assumes that the domain knowledge $\hat{r}$ is given by humans while the domain knowledge $\hat{r}$ is learned in our case. Moreover, paper (Hu et al., 2020) assumes that the shaping reward is linear, i.e., the shaping reward is $r(s, a) + \theta(s, a)(r(s, a) - \hat{r}(s, a))$. In contrast, our shaping reward class $r_\theta(r(s, a), r(s, a) - \hat{r}(s, a))$ is more general and includes the linear shaping reward as a special case.

Algorithm: Paper (Hu et al., 2020) proposes several methods to compute the hyper-gradient (i.e., the gradient of the upper-level problem). However, they do not consider the practical issue, i.e., the lower-level problem cannot be fully solved in finite time. In contrast, our constrained bi-level optimization problem cannot be solved by the state-of-the-arts (Xu & Zhu, 2023; Khanduri et al., 2023) so that we develop a novel algorithm that solve the problem and we consider the practical issue, i.e., we cannot obtain the exact optimal solution in finite time.

Theoretical analysis: Paper (Hu et al., 2020) does not have theoretical guarantees at all. In contrast, we propose a systematic theoretical framework, which is one of our core contributions.

F Experiment details

The code was running on a laptop whose CPU is Intel Core i9 12900k and GPU is NVIDIA RTX 3080. The operating system is Windows 10. We use a neural network to parameterize the learned reward function. The neural network has two hidden layers where each hidden layer has 64 neurons. The activation functions are respectively ReLU and Tanh.

The delayed MuJoCo environments. The delayed Mujoco environments are widely used in RL improvement literature (Zheng et al., 2018; Memarian et al., 2021; Oh et al., 2018) where the reward is accumulated by 20 time steps and only provided at the end. For example, for an episode with length 100 (i.e., $0, \cdots, 99$), only the time steps 19, 39, 59, 79, 99 receive nonzero reward while all the other steps receive zero reward. The time step 19 receives the reward that is accumulated from time 0 to time 19. We can see that the delayed MuJoCo tasks have sparse reward.

F.1 Using other baseline RL algorithms

We provide the results of using PPO and TD3 as the baseline RL algorithm below:

Table 4: Using PPO as the baseline RL algorithm.

	Delayed HalfCheetah	Delayed Hopper	Delayed Walker	Delayed Ant	HalfCheetah	Hopper	Walker	Ant
PPO	179.97 ± 20.12	243.14 ± 25.87	175.36 ± 18.22	33.28 ± 5.10	221.58 ± 25.29	285.10 ± 36.29	203.19 ± 24.84	302.94 ± 28.29
UTILITY	422.83 ± 32.15	400.99 ± 21.45	395.15 ± 30.28	88.42 ± 5.54	457.03 ± 30.29	420.46 ± 35.94	402.40 ± 39.28	392.95 ± 39.10
RICE	226.54 ± 16.35	271.85 ± 18.73	205.15 ± 16.11	42.62 ± 3.48	238.17 ± 19.99	295.82 ± 21.18	229.52 ± 23.22	315.38 ± 22.38
SIL	232.55 ± 25.75	259.15 ± 22.43	183.27 ± 10.39	49.66 ± 3.21	271.92 ± 29.18	301.93 ± 28.37	284.44 ± 48.17	311.92 ± 24.73
LIR	341.38 ± 22.54	311.44 ± 21.18	315.77 ± 28.59	73.14 ± 6.28	319.29 ± 37.19	314.73 ± 30.18	326.28 ± 28.27	325.83 ± 18.37

Table 5: Using TD3 as the baseline RL algorithm.

	Delayed HalfCheetah	Delayed Hopper	Delayed Walker	Delayed Ant	HalfCheetah	Hopper	Walker	Ant
TD3	216.84 ± 18.50	252.84 ± 19.68	167.02 ± 21.52	43.15 ± 4.28	237.29 ± 22.51	283.75 ± 28.14	193.25 ± 22.15	364.82 ± 21.17
UTILITY	464.74 ± 27.06	422.13 ± 24.15	392.04 ± 24.77	105.24 ± 7.88	476.92 ± 27.15	433.15 ± 27.13	396.16 ± 28.59	483.11 ± 26.85
RICE	221.16 ± 16.88	269.17 ± 14.66	179.13 ± 12.83	46.09 ± 3.83	236.83 ± 19.26	289.99 ± 21.77	212.26 ± 21.04	371.65 ± 22.85
SIL	258.16 ± 23.92	308.63 ± 26.25	201.66 ± 12.95	58.62 ± 6.38	312.29 ± 24.88	315.17 ± 24.72	289.51 ± 31.43	336.16 ± 21.63
LIR	377.19 ± 26.56	346.10 ± 18.11	298.17 ± 23.77	88.19 ± 3.64	386.10 ± 33.83	366.28 ± 26.88	335.15 ± 20.62	381.17 ± 32.15

Table 4 and Table 5 show that UTILITY significantly outperforms the baselines when the baseline RL algorithms are PPO and TD3.

F.2 Experiment results on dense reward

To show that our method can also improve the performance on dense reward scenarios, we include the experiment results on the original MuJoCo environment (where the reward is dense) below:

Table 6: Experiment results (original MuJoCo environment with dense reward).

	SAC	UITLITY	RICE	SIL	LIR
HalfCheetah	686.40 ± 41.24	824.42 ± 42.18	701.44 ± 45.83	716.62 ± 39.17	718.25 ± 42.10
Hopper	238.14 ± 29.94	348.16 ± 26.32	242.99 ± 19.10	264.86 ± 27.11	277.58 ± 22.69
Walker2d	182.21 ± 24.14	269.14 ± 25.08	189.19 ± 39.11	197.63 ± 21.39	214.16 ± 21.24
Ant	299.79 ± 26.53	421.63 ± 25.16	322.26 ± 34.15	340.14 ± 26.53	351.14 ± 28.92

Table 6 shows the final results on the original MuJoCo environment (dense reward). We can observe that UTILITY achieves the highest reward and largely improves SAC.

F.3 Experiment results when the episode length is 1000

We provide the experiment results when the episode length is $1,000$.

Table 7: Results when the episode length is $1,000$.

	Delayed HalfCheetah	Delayed Hopper	Delayed Walker	Delayed Ant	HalfCheetah	Hopper	Walker	Ant
SAC	3019.18 ± 88.69	1975.28 ± 66.10	1827.43 ± 108.19	3652.85 ± 59.70	5021.39 ± 102.15	3592.29 ± 88.62	3344.21 ± 95.58	5922.10 ± 128.19
UTILITY	4872.10 ± 86.39	3562.99 ± 74.44	3262.52 ± 79.01	4929.14 ± 74.26	5806.28 ± 69.01	4217.83 ± 75.62	3829.16 ± 73.07	6500.73 ± 96.18
RICE	3138.44 ± 35.66	2196.82 ± 64.29	1996.53 ± 50.82	3742.18 ± 49.45	5127.77 ± 42.72	3719.42 ± 63.17	3396.28 ± 50.27	5998.26 ± 62.16
SIL	3528.53 ± 56.03	2549.98 ± 85.43	2393.75 ± 56.56	3940.95 ± 60.10	5429.58 ± 74.95	3692.48 ± 35.84	3462.94 ± 44.82	6102.58 ± 57.29
LIR	3921.61 ± 142.06	2842.19 ± 71.02	2511.26 ± 38.51	4020.29 ± 61.02	5420.92 ± 59.29	3792.40 ± 101.93	3502.94 ± 24.64	6204.56 ± 39.12

Table 7 shows that UTILITY outperforms the baselines when the episode length is 1,000.

F.4 Ablation study

Since our method has two components to improve the performance: the shaping reward and the constrained formulated by the "misleading" state-action pairs. Here, we separately study the effect of the learned shaping reward and the constraint. In specific, we test the performance of the shaping only method and the constraint only method, and provide the results below:

Table 8: Ablation study for dense reward.

	SAC	UITLITY	shaping only	constraint only
HalfCheetah	686.40 ± 51.24	824.42 ± 42.18	764.25 ± 48.11	701.19 ± 47.43
Hopper	238.14 ± 29.94	348.16 ± 26.32	311.78 ± 34.24	268.15 ± 42.66
Walker2d	182.21 ± 24.14	269.14 ± 25.08	242.18 ± 29.62	196.77 ± 22.19
Ant	299.79 ± 26.53	421.63 ± 25.16	396.05 ± 29.18	317.12 ± 34.59

Table 8 shows that both the learned shaping reward and the constraint can improve the performance, and the shaping reward has a larger impact. Moreover, even if we only use the shaping reward, the performance is better than LIR. This is because our shaping reward uses the domain knowledge formulated by the high-level explanation. Even if we only use the constraint, the performance is better than RICE. The reason is that the "misleading" state-action pairs we find are the points that lead to the failure, and thus avoiding these state-action pairs can improve performance. In contrast, RICE finds the states that are most influential cumulative reward, however, these states may not be the states that lead the RL agent to be non-optimal.

F.5 Fidelity of the generated two-level explanation

The fidelity means the correctness of the two-level explanation (Guo et al., 2021b; Cheng et al., 2023). Since the two-level explanation is to explain why the RL agent (i.e., SAC) is not optimal, one way to validate the fidelity of the explanation is to see whether the performance improves after we improve from the explanations. From the last two columns in Table 8, we can see that both the high-level and low-level explanations are the correct explanations because both the shaping only method and the constraint only method improve the performance. Moreover, the shaping only method (the fourth column in Table 8) has a higher cumulative reward than LIR (the last column in Table 6), and the constraint only method (the last column in Table 8) has a higher cumulative reward than RICE (the fourth column in Table 6). This shows the high fidelity of our two-level explanation.

Table 9: Fidelity comparison for dense reward.

	SAC	shaping only (ours)	shaping without $r - \hat{r}$	constraint only (ours)	RICE+constraint
HalfCheetah	686.40 ± 51.24	764.25 ± 48.11	710.99 ± 35.16	701.19 ± 47.43	672.15 ± 25.16
Hopper	238.14 ± 29.94	311.78 ± 34.24	279.12 ± 27.44	268.15 ± 42.66	211.03 ± 21.20
Walker2d	182.21 ± 24.14	242.18 ± 29.62	213.15 ± 18.94	196.77 ± 22.19	177.19 ± 15.33
Ant	299.79 ± 26.53	396.05 ± 29.18	359.14 ± 22.85	317.12 ± 34.59	288.19 ± 18.02

Table 9 shows that both the high-level and low-level explanations of our method have higher fidelity. The method "RICE+constraint" has even worse performance than SAC because the critical states influential to the cumulative reward may be the states that lead to high cumulative reward, and thus constraining them may even make the performance worse. However, even if we do not constrain these states but use the fine-tune method as in (Cheng et al., 2023) instead, our constraint-only method (the last column in Table 8) still outperforms RICE (the fourth column in Table 6). For the high-level explanation, we can see that our shaping only method achieves higher cumulative reward than the method "shaping without $r - \hat{r}$". This shows the high fidelity of our high-level explanation.

F.6 How to accelerate the triple-loop algorithm

The total iterations of UTILITY is $N \times \bar{N} \times \tilde{N}$ where N is the iteration number of the outer loop, $\bar{N}$ is the iteration number of the middle loop, and $\tilde{N}$ is the iteration number of the inner loop. While the triple-loop structure looks computationally expensive, in practice, we can significantly accelerate the algorithm using warm start in the inner loop and middle loop. Take the inner loop as an example, given the shaping parameter θ and the current dual parameter $\lambda_{\bar{n}}$, we need to compute the corresponding constrained soft Bellman policy $\pi_{\lambda_{\bar{n}};\theta}$ in the inner loop. Instead of starting from a random policy initialization, we use the policy $\hat{\pi}_{\lambda_{\bar{n}-1};\theta}$ learned in last inner loop as the initialization where $\hat{\pi}_{\lambda_{\bar{n}-1};\theta}$ is an approximation of $\pi_{\lambda_{\bar{n}-1};\theta}$. The intuition behind this is that since $\lambda_{\bar{n}}$ and $\lambda_{\bar{n}-1}$ are close (only different by one-step gradient descent), it is expected that $\pi_{\lambda_{\bar{n}-1};\theta}$ and $\pi_{\lambda_{\bar{n}};\theta}$ are close. Therefore, using $\hat{\pi}_{\lambda_{\bar{n}-1};\theta}$ as the initialization makes it easier to approach $\pi_{\lambda_{\bar{n}};\theta}$. Therefore, the warm start trick enables us to use fewer iterations for the inner loop, i.e., the iteration number $\tilde{N}$ reduces. We use the similar warm start trick for the middle loop to reduce the iteration number $\bar{N}$.

Empirically, in our experiment, we use warm start and set inner iteration number $\tilde{N} = 1$ and middle iteration number $\bar{N} = 2$. We can see that even if it is a triple-loop algorithm, the total iteration number is small, i.e., $2N$. This warm start trick is inspired by (Zeng et al., 2022; Liu & Zhu, 2024a) where they also use warm start and only run the inner loop for one iteration and the final results are not worse than the algorithm that runs the inner loop for many iterations starting from random initialization. We run the code on a desktop whose CPU is Intel Core i9 12900k and GPU is NVIDIA RTX 3080. We include the runtime of our UTILITY algorithm below:

Table 10: Computation time.

	Delayed HalfCheetah	Delayed Hopper	Delayed Walker	Delayed Ant	HalfCheetah	Hopper	Walker	Ant
UTILITY	2h30min	2h10min	2h32min	3h11min	2h26min	2h5min	2h34min	3h09min
RICE	2h18min	1h53min	2h19min	2h55min	2h18min	1h59min	2h10min	2h58min
SIL	3h14min	3h32min	3h58min	4h43min	3h55min	3h8min	4h11min	4h49min
LIR	2h41min	2h5min	2h29min	3h15min	2h35min	2h13min	2h45min	3h16min

Table 10 shows that the computation time of UTILITY is comparable to the baselines due to the warm start trick.

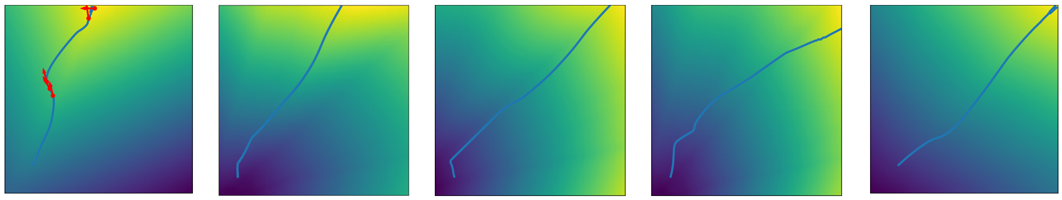
G Potential societal impact

This paper has positive impact which is to improve the performance of RL agents. However, the paper also has potential negative impacts. Since the two-level explanation identifies the mistakes made by the RL agents. A malicious entity may use these weaknesses or mistakes to launch attack to the RL agents. To alleviate this issue, one solution is to keep the demonstration of the RL agent private, so that the malicious entity cannot get access to the demonstration and thus cannot find the weakness.

H Limitations

One limitation of the method is that it requires to interact with the environment. Therefore, one future work is to extend this method to the offline RL setting.

I Evolution fro Figure 1c to Figure 1d



(a) Iteration=0 (b) Iteration=50 (c) Iteration=100 (d) Iteration=150 (e) Iteration=200

Figure 3: Evolution from Figure 1c to Figure 1d.

For this scenario, we run UTILITY for 200 iterations. Figure 3 show the evolution from Figure 1c (iteration=0) to Figure 1d (iteration=200). From the evolution, we can see that the learned reward becomes closer and closer to the ground truth reward and the learned policy becomes more and more optimal.