Reward-Estimated Hypergradient for Bilevel Reinforcement Learning with Black-Box Follower

Shigeki Kusaka[*†‡] **Mikoto Kudo**[*†] **Takumi Tanabe**[§]

Akifumi Wachi[§] **Youhei Akimoto**[*†‡]

[*]University of Tsukuba [†]RIKEN AIP [‡]Institute of Science Tokyo [§]LY Corporation
{shigeki.kusaka, mikoto}@bbo.cs.tsukuba.ac.jp
{akumi.tanabe, akifumi.wachi}@lycorp.co.jp
akimoto@cs.tsukuba.ac.jp

Abstract

Hypergradient (HG) methods for bilevel reinforcement learning (RL) typically assume a white-box follower whose reward is known. In practice, however, the leader rarely knows the follower's reward function and can only observe its trajectories. We address this black-box setting and propose Reward-Estimated Hypergradients (RE-HG), which recovers the follower's reward from these trajectories via inverse RL (IRL). Yet IRL identifies the reward only up to potential-based shaping, a transformation that leaves the follower's behavior unchanged but biases the HG. We prove that, for linear rewards, this bias cancels when IRL estimates are aggregated across sufficiently diverse dynamics, which the leader's own updates naturally supply. To handle finite samples, RE-HG truncates small eigenvalues of the aggregated feature covariance, restoring the identifiable subspace and stabilizing the estimation. Experiments show that RE-HG estimates gradients aligned with a white-box Oracle and achieves comparable performance; empirically, the truncation also suppresses HG explosions, letting RE-HG avoid local optima that trap the Oracle in sharp reward landscapes.

1 Introduction

Reinforcement Learning (RL) (Sutton and Barto, 2018) has achieved remarkable success in games (Silver et al., 2016; Samvelyan et al., 2019), robotics (Levine et al., 2016), and the fine-tuning of large language models (Ouyang et al., 2022; Guo et al., 2025). Many real-world deployments, however, involve multiple agents whose interaction is hierarchical: one agent, the *leader*, shapes the environment in which another agent, the *follower*, learns. Bilevel RL (Hong et al., 2023) models this structure as a nested optimization problem, in which the leader optimizes its objective while anticipating the follower's best response:

$$\max_{\theta\in\Theta} J_L(\theta, g^{\theta\dagger}) \qquad \text{(Leader)}$$

$$\text{s.t. } g^{\theta\dagger} = \operatorname*{argmax}_{g\in\mathcal{G}} J_\theta(g). \qquad \text{(Follower)}$$

Here, θ parameterizes the leader, J_L is the leader's objective, and $g^{\theta\dagger}$ is the follower's best response, which maximizes the follower's own objective J_θ defined by its reward function. The standard approach to optimizing the leader is to compute the hypergradient (HG), the gradient of J_L with respect to θ that accounts for how the follower's best response $g^{\theta\dagger}$ changes with θ.

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

Existing HG methods, however, assume a white-box follower: they require the follower's reward function and, in some cases, its policy or value function (Table 1). In practice, this assumption rarely holds: the follower's preferences are often implicit, as with human arm movements (Albrecht et al., 2012) or human drivers in mixed traffic (Shi et al., 2025), and manually engineering a reward function that reflects them is notoriously difficult. To the best of our knowledge, no established method computes the HG in this black-box follower setting, where the leader can observe only the follower's trajectories.

A natural approach is to recover the follower's reward from its trajectories by inverse RL (IRL) and to compute the HG with the recovered reward. However, IRL identifies the reward only up to potential-based shaping (Ng et al., 1999): shaped rewards induce the same optimal policy, so the follower's trajectories cannot distinguish between them. In single-agent RL, this ambiguity is harmless, since shaping adds no bias to the policy gradient. The HG, in contrast, depends directly on the follower's value function, which shaping shifts by an arbitrary potential, so the ambiguity left by IRL becomes an arbitrary bias in the HG.

Our key observation is that the bilevel structure itself supplies the information needed to resolve this ambiguity. Prior work has shown that a reward can be identified up to a constant when expert behavior is observed under multiple transition dynamics (Cao et al., 2021; Rolland et al., 2022). In bilevel RL, such observations arise naturally: each leader update changes the dynamics that the follower faces, and the follower responds with a new best response. We show that, under a linear reward assumption, aggregating the IRL estimates obtained across these iterations cancels the shaping bias and recovers the reward up to a constant offset, which leaves the HG unchanged. To make this aggregation reliable with finite samples, we truncate the small eigenvalues of the aggregated feature covariance, which restores the identifiable subspace and keeps the estimation numerically stable. Combining these components, we propose *Reward-Estimated Hypergradient* (RE-HG), which computes the HG from observed trajectories alone.

The main contributions of this work are as follows:

(i) We identify reward-shaping ambiguity as the fundamental obstacle to HG estimation with a black-box follower.

(ii) We prove that, under a linear reward assumption, aggregating IRL estimates across the dynamics induced by the leader's updates cancels this bias (Theorem 1), and that the remaining reward error controls the HG error (Proposition 2).

(iii) We propose RE-HG, which combines this aggregation with eigenvalue truncation to compute the HG from trajectories alone with finite samples.

(iv) Experiments on discrete and continuous bilevel tasks show that RE-HG estimates gradients aligned with a white-box ORACLE and achieves comparable performance. Empirically, eigenvalue truncation also suppresses HG explosions, letting RE-HG avoid local optima that trap the ORACLE in sharp reward landscapes.

Our code is available at https://github.com/akimotolab/RE-HG.

2 Problem Formulation

Entropy Regularized MDPs In this work, we assume that a follower solves a Markov decision process (MDP) with entropy regularization. An MDP is defined by the tuple $\langle \mathcal{S}, \mathcal{B}, p, \rho_0, r_F, \gamma_F \rangle$, where $\mathcal{S}$ represents the state space, $\mathcal{B}$ denotes the follower's action space, and $p : \mathcal{S} \times \mathcal{B} \times \mathcal{S} \to [0, 1]$ is the state transition probability. The initial state distribution is given by $\rho_0 : \mathcal{S} \to [0, 1]$, the follower's reward function is $r_F : \mathcal{S} \times \mathcal{B} \to \mathbb{R}$, and $\gamma_F \in [0, 1)$ is the discount factor.

The follower aims to identify a stochastic policy $g \in \mathcal{G} : \mathcal{S} \times \mathcal{B} \to [0, 1]$ that maximizes the expected discounted return plus an entropy bonus:

$$\max_{g \in \mathcal{G}} J(g) := \mathbb{E}_\tau \left[\sum_{t=0}^{\infty} \gamma_F^t \left(r_F(s_t, b_t) + \beta H(g; s_t) \right) \right], \tag{1}$$

where $\mathcal{G}$ is the set of all valid policies on $\mathcal{S} \times \mathcal{B}$, and $\beta > 0$ determines the regularization strength. The term $H(g; s) := -\mathbb{E}_{b \sim g(\cdot|s)}[\log g(b \mid s)]$ represents the entropy of policy g at state $s \in \mathcal{S}$. The

expectation is taken over trajectories $\tau := (s_0, b_0, s_1, b_1, \dots)$ distributed according to $p(\tau \mid g) = \rho_0(s_0) \prod_{t=0}^{\infty} p(s_{t+1} \mid s_t, b_t) g(b_t \mid s_t)$.

The primary motivation for employing entropy regularization is to ensure that the optimal policy is both unique and stochastic. This framework is essential in inverse reinforcement learning (IRL), where it serves as a natural mechanism for modeling the inherent stochasticity of expert behavior. Under this setting, we define the "soft" value function V_F^g and action-value function Q_F^g as:

$$V_F^g(s) := \mathbb{E}_\tau \left[\sum_{t=0}^{\infty} \gamma_F^t \left(r_F(s_t, b_t) + \beta H(g; s_t) \right) \mid s_0 = s \right]$$

and $Q_F^g(s, b) := r_F(s, b) + \gamma_F \mathbb{E}_{s' \sim p(\cdot \mid s, b)}[V_F^g(s')]$. Given a sufficiently expressive policy class, the unique optimal policy $g^\star$ is characterized by the Boltzmann distribution relative to the optimal value functions $V_F^\star(s) = \max_{g \in \mathcal{G}} V_F^g(s)$ and $Q_F^\star(s, b) = \max_{g \in \mathcal{G}} Q_F^g(s, b)$ (Haarnoja et al., 2017):

$$g^\star(b \mid s) = \exp\left(\beta^{-1} \left(Q_F^\star(s, b) - V_F^\star(s) \right) \right).$$

This uniqueness is a critical property for the stability and well-definedness of bi-level RL problems.

Configurable Markov Decision Process A Configurable Markov Decision Process (Conf-MDP) extends the standard MDP by allowing certain environmental dynamics to be tuned via a configuration parameter. We denote a Conf-MDP as $\mathcal{M}_\theta$, where $\theta \in \Theta$ parameterizes the transition probability p^θ and the initial state distribution ρ_0^θ. Thus, the environment is defined by the tuple $\langle \mathcal{S}, \mathcal{B}, p^\theta, \rho_0^\theta, r_F, \gamma_F \rangle$. For any fixed configuration $\theta \in \Theta$, the follower solves the entropy-regularized objective J_θ defined by (1) using the parameterized dynamics p^θ and ρ_0^θ.

The leader's goal is to optimize their own utility by selecting a configuration θ that accounts for the follower's subsequent best response. This results in the following bi-level optimization problem:

$$\max_{\theta \in \Theta} J_L(\theta, g^{\theta\dagger}), \quad \text{where} \quad g^{\theta\dagger} := \operatorname*{argmax}_{g \in \mathcal{G}} J_\theta(g). \tag{2}$$

Here, J_L is the leader's utility and $g^{\theta\dagger}$ is the follower's best response to θ. Because of the entropy regularization, $g^{\theta\dagger}$ is unique for every θ and is given by the θ-dependent Boltzmann distribution:

$$g^{\theta\dagger}(b \mid s) = \exp\left(\frac{Q_F^{\theta\dagger}(s, b) - V_F^{\theta\dagger}(s)}{\beta} \right),$$

where $Q_F^{\theta\dagger}$ and $V_F^{\theta\dagger}$ are the optimal soft value functions. We assume the follower always reacts with its optimal policy. The leader's utility is defined as the expected discounted cumulative reward:

$$J_L(\theta, g^{\theta\dagger}) := \mathbb{E}_\tau \left[\sum_{t=0}^{\infty} \gamma_L^t r_L^\theta(s_t, b_t) \right], \tag{3}$$

where γ_L is the leader's discount factor and r_L^θ is the leader's reward function. Because the leader has control over the environment, we assume that the leader knows p^θ and ρ_0^θ. Moreover, we assume that the gradients $\nabla_\theta r_L^\theta$, $\nabla_\theta \log \rho_0^\theta$, and $\nabla_\theta \log p^\theta$ exists and the leader can compute them explicitly.

2-Player Markov Game A 2-player Markov Game (MG) further generalizes the Conf-MDP by treating the leader as an active agent. In this setting, the leader follows a policy f_θ parameterized by $\theta \in \Theta$, indirectly influencing the environment through their actions. An MG is defined by the tuple $\langle \mathcal{S}, (\mathcal{A}, \mathcal{B}), p, \rho_0, (r_L, r_F), (\gamma_L, \gamma_F) \rangle$, where $\mathcal{A}$ and $\mathcal{B}$ are the action spaces for the leader and follower, respectively. Transitions $p(s' \mid s, a, b)$ and rewards r_L, r_F now depend on both agents' actions. The policies are $f_\theta : \mathcal{S} \times \mathcal{A} \to [0, 1]$ and $g : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to [0, 1]$.

As in the Conf-MDP, the follower optimizes the entropy-regularized objective (1), but the dynamics are now induced by the leader's policy: $p^\theta(s', a' \mid s, a, b) = f_\theta(a' \mid s') p(s' \mid s, a, b)$. The leader's optimization remains consistent with (2), with the utility defined as:

$$J_L(\theta, g^{\theta\dagger}) := \mathbb{E}_\tau \left[\sum_{t=0}^{\infty} \gamma_L^t r_L(s_t, a_t, b_t) \right], \tag{4}$$

where $\tau = (s_0, a_0, b_0, s_1, \dots)$ consists of state, leader action, and follower action. We define the follower's soft value and action-value functions under a specific leader policy f_θ as:

$$V_F^{\theta,g}(s,a) := \mathbb{E}_\tau \left[\sum_{t=0}^{\infty} \gamma_F^t \left(r_F(s_t, a_t, b_t) + \beta H(g; s_t, a_t) \right) \middle| \begin{array}{l} s_0 = s \\ a_0 = a \end{array} \right]$$

and $Q_F^{\theta,g}(s,a,b) := r_F(s,a,b) + \gamma_F \mathbb{E}_{s' \sim p} \mathbb{E}_{a' \sim f_\theta}[V_F^{\theta,g}(s', a')]$. Here, $V_F^{\theta,g}$ takes the leader's action a as an input to reflect the follower's observation of the leader's move. The leader's action-value function is defined without entropy regularization:

$$Q_L^{\theta,g}(s,a,b) := \mathbb{E}_\tau \left[\sum_{t=0}^{\infty} \gamma_L^t r_L(s_t, a_t, b_t) \middle| \begin{array}{l} s_0 = s \\ a_0 = a \\ b_0 = b \end{array} \right].$$

For brevity, we use the notation $Q_L^{\theta\dagger}$, $V_F^{\theta\dagger}$, and $Q_F^{\theta\dagger}$ to denote the respective value functions when the follower plays their unique best response $g^{\theta\dagger}$ to the leader's policy f_θ. Unlike Conf-MDP settings, the leader may not know the transition probability. We consider two scenarios: the leader (1) knows p and ρ_0 explicitly; and (2) does not know them explicitly, but can simulate the interaction with p and ρ_0.

Inverse Reinforcement Learning Inverse Reinforcement Learning (IRL) is the process of recovering an underlying reward function from an observed policy, assuming that the policy is optimal with respect to that reward. In our framework, we employ the Maximum Discounted Causal Entropy (MDCE) IRL framework (Zhou et al., 2018) to estimate the follower's reward function based on their best response $g^{\theta\dagger}$ under a given leader policy f_θ. To formalize this, let $J_\theta(g; r)$ represent the follower's objective as defined in (1), but with the true reward r_F replaced by a surrogate reward r, and the regularization constant set to $\beta = 1$. This simplification is without loss of generality, as the scaling factor β is effectively absorbed into the surrogate reward; specifically, maximizing $J_\theta(g; r_F/\beta)$ with $\beta = 1$ is mathematically equivalent to the original formulation in (1). We denote the solution to the forward reinforcement learning problem under a surrogate reward r as: $\mathrm{RL}_\theta(r) = \operatorname{argmax}_g J_\theta(g; r)$. Given the follower's best response $g^{\theta\dagger}$ as the "expert" policy, the IRL problem is formulated as finding a reward function r that minimizes the difference between the maximum achievable value and the value attained by the expert:

$$\mathrm{IRL}_\theta(g^{\theta\dagger}) = \operatorname*{argmin}_r \left(\max_g J_\theta(g; r) - \mathbb{E}_{\tau \sim (f_\theta, g^{\theta\dagger})} \left[\sum_{t=0}^{\infty} \gamma_F^t r(s_t, a_t, b_t) \right] \right),$$

where the expectation is taken over trajectories τ generated by the leader f_θ and the follower $g^{\theta\dagger}$. As established in Ho and Ermon (2016, Corollary 3.2.1) and Syed et al. (2008, Theorem 2), if a policy $\hat{g}$ is obtained such that $\hat{g} \in \mathrm{RL}_\theta \circ \mathrm{IRL}_\theta(g^{\theta\dagger})$, then we successfully recover the follower's behavior, i.e., $\hat{g} = g^{\theta\dagger}$. This implies that the reward function $\hat{r}$ resulting from the IRL process effectively characterizes the optimal behavior of the follower under the leader parameter θ.

3 Related Work

Hypergradient for Bilevel RL In Bilevel RL, computing the HG to optimize the leader's policy has become a standard approach. While existing methods have evolved significantly in terms of convergence rates and the relaxation of theoretical assumptions, they ultimately rely on the "white-box" nature of the follower to varying degrees. For instance, PARL (Chakraborty et al., 2024) and PBRL (Shen et al., 2024) assume deterministic updates that require full knowledge of the HG, while HPGD (Thoma et al., 2024) and SoBiRL (Yang et al., 2025) are bottlenecked by the requirement for a sampler capable of starting from arbitrary state-action pairs or allowing repeated visits to those pairs. Although BC-HG circumvents the need for repeated visits, it still necessitates direct access to the follower's policy and value functions. As summarized in Table 1, all these approaches, regardless of their estimation technique, rely on explicit knowledge of the follower's true reward function, which RE-HG does not require.

Table 1: Comparison of requirements for HG estimation methods in Bilevel RL.

Method	Requires Follower's True Reward	Update Method (HG Info)	Requires Hessian (2nd Order Deriv.)	Requires Repeated State Visits
PARL (Chakraborty et al., 2024)	✓	Deterministic	✓	✗
PBRL (Shen et al., 2024)	✓	Deterministic	✗	✗
SoBiRL (Yang et al., 2025)	✓	Stochastic	✗	✓
HPGD (Thoma et al., 2024)	✓	Stochastic	✗	✓
BC-HG (Kudo et al., 2026)	✓	Stochastic	✗	✗
RE-HG (Ours)	✗	Stochastic	✗	✗

Inverse Reinforcement Learning While the optimal policy for a given reward function is uniquely identified under entropy regularization, the reward function leading to the given policy is not uniquely determined. Ng et al. (1999) shows that for an arbitrary function $\Phi : \mathcal{S} \to \mathbb{R}$, the shaped reward function $r'(s, b) = r(s, b) + \gamma_F \mathbb{E}_{s' \sim p(\cdot|s,b)}[\Phi(s')] - \Phi(s)$ admits the optimal policy equivalent to that for the original reward function r. It has been argued that a shaped reward function is insufficient to obtain the optimal policy when the transition probability used for the policy training is different from the one used for the expert trajectory generation. Several works propose approaches to recover the original reward function (up to a constant shift, which is unavoidable) under different assumptions. For example, Fu et al. (2018); Ni et al. (2020) assume a state-only reward, along with other conditions on the transition. Cao et al. (2021); Rolland et al. (2022) assume access to the expert trajectories on two different MDPs sharing the same reward function. We develop a method based on the latter idea. A direct comparison against the joint multi-environment formulation of Cao et al. (2021) is provided in Section C.7.

4 Approach

We derive a novel estimation of the HG of the leader's objective. As a baseline, we employ BC-HG (Kudo et al., 2026) for the HG estimate that assumes the knowledge of the optimal value function of the follower. Our approach, RE-HG, incorporates an inverse reinforcement learning technique to mitigate the necessity of the optimal value function of the follower.

Baseline: Hypergradient estimation with Known Follower's Value If the follower's optimal value function is accessible, the HG of the leader's objective can be computed both for Conf-MDP and 2-player MG settings. The HG for the Conf-MDP setting is derived as follows. Let $Q_L^{\theta,g}(s, b)$ represent the leader's expected discounted return given that the follower chooses action b at state s:

$$Q_L^{\theta g}(s, b) := \mathbb{E}_\tau \left[\sum_{t=0}^{\infty} \gamma_L^t r_L(s_t, b_t) \,\middle|\, \begin{array}{l} s_0 = s \\ b_0 = b \end{array} \right].$$

Distinct from standard action-value functions, this definition treats the follower's action b as the primary input alongside the state s, rather than the leader's action. We denote the leader's value function under the follower's optimal response as $Q_L^{\theta\dagger} := Q_L^{\theta, g^{\theta\dagger}}$. Then, a concept of the *benefit* is introduced. For a given state s, the benefit $B_L^{\theta\dagger}(s, b)$ quantifies the relative advantage the leader gains when the follower selects action b compared to the follower's average behavior under $g^{\theta\dagger}$:

$$B_L^{\theta\dagger}(s, b) := Q_L^{\theta\dagger}(s, b) - \mathbb{E}_{b' \sim g^{\theta\dagger}(\cdot|s)} \left[Q_L^{\theta\dagger}(s, b') \right].$$

Kudo et al. (2026) show that the HG of J_L in (3) can be computed in the case of Conf-MDPs as

$$\nabla_\theta J_L(\theta, g^{\theta\dagger}) = \frac{\partial J_L(\theta, g^{\theta\dagger})}{\partial \theta} + \mathbb{E}_\tau \left[\sum_{t=0}^{\infty} \gamma_L^t \frac{B_L^{\theta\dagger}(s_t, b_t)}{\beta} \nabla_\theta Q_F^{\theta\dagger}(s_t, b_t) \right], \tag{5}$$

where $\frac{\partial J_L(\theta, g^{\theta\dagger})}{\partial \theta}$ is the partial derivative of $J_L(\theta, g)$ with respect to θ evaluated at $(\theta, g^{\theta\dagger})$, whose explicit form is provided in Section B.1. The gradient of the follower's soft action-value function is:

$$\nabla_\theta Q_F^{\theta\dagger}(s, b) = \mathbb{E}_\tau \left[\sum_{t=0}^{\infty} \gamma_F^{t+1} V_F^{\theta\dagger}(s_{t+1}) \nabla_\theta \log p^\theta(s_{t+1} \mid s_t, b_t) \,\middle|\, \begin{array}{l} s_0 = s \\ b_0 = b \end{array} \right]. \tag{6}$$

The HG can be computed similarly in 2-player MGs, described in Section B.1. Importantly, in both cases, the term $\frac{B_L}{\beta}\nabla_\theta Q_F$ appears. We refer to it as the *influence term*, as it accounts for the change in the follower's best response induced by the leader's configuration.

Follower's Value Estimation via Repeated Inverse Reinforcement Learning The challenge in estimating the HG without knowing the follower's reward function is to estimate the value function of the follower, $V_F^{\theta\dagger}$. As mentioned earlier, an IRL returns a reward function $\hat{r} \in \mathrm{IRL}_\theta(g^{\theta\dagger})$ that admits the same optimal policy $g^{\theta\dagger}$. In Conf-MDP settings, it includes any function $r_\Phi(s,b) = r(s,b) + \gamma_F \mathbb{E}[\Phi(s') \mid s,b] - \Phi(s)$ with an arbitrary state-only potential function $\Phi : \mathcal{S} \to \mathbb{R}$ (Ng et al., 1999). The soft value function under the shaped reward r_Φ is then $V_\Phi^{\theta\dagger} = V_F^{\theta\dagger} - \Phi$. Therefore, if we use a shaped reward to estimate the follower's value function, the term $\nabla_\theta Q_F^{\theta\dagger}$ will be arbitrarily affected by Φ, resulting in biased estimation of the HG $\nabla_\theta J_L(\theta, g^{\theta\dagger})$. The same argument holds for 2-player MG settings. In this case, the leader's action is regarded as a part of the state from the follower's perspective. Hence, the estimated reward is shaped as $r_\Phi(s,a,b) = r(s,a,b) + \gamma_F \mathbb{E}[\Phi(s',a') \mid s,a,b] - \Phi(s,a)$ with an arbitrary potential function $\Phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. Consequently, the follower's value function is misestimated as $V_\Phi^{\theta\dagger} = V_F^{\theta\dagger} - \Phi$, yielding a biased HG due to the additional term $-\mathbb{E}_\tau\left[\sum \gamma_F^{t+1}\Phi(s_{t+1})\nabla_\theta \log p^\theta(s_{t+1} \mid s_t, b_t)\right]$ arising from substituting $V_\Phi^{\theta\dagger}$ in (6). Different from the single-agent MDPs, where the reward shaping adds no bias in policy gradient, the shaping term is crucial for HG. The defect of the shaping term is demonstrated in Section 5 as well as in Section C.2.

To circumvent this hurdle, we pose the assumption on the follower's reward function. In what follows, we consider Conf-MDP settings for conciseness. To treat 2-player MGs, we can simply regard a state $s \in \mathcal{S}$ as a pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ of the state and the leader's action at state s, the transition probability as $p(s' \mid s,a,b) f_\theta(a' \mid s')$, and the initial state distribution as $\rho_0(s) f_\theta(a \mid s)$.

Assumption 1 (Follower's Reward Linearity)**.** *For a given feature extractor $\phi : \mathcal{S} \times \mathcal{B} \to \mathbb{R}^d$, the follower's reward function is given by $r_F(s,b) = \langle \omega^\star, \phi(s,b) \rangle$ with $\omega^\star \in \mathbb{R}^d$. Moreover, the features have full rank, i.e.,* $\textsc{Span}(\{\phi(s,b) : (s,b) \in \mathcal{S} \times \mathcal{B}\}) = \mathbb{R}^d$.

Reward linearity is not sufficient to identify the true reward function up to a constant offset (Cao et al., 2021; Rolland et al., 2022). However, the reward parameter obtained by $\mathrm{IRL}_\theta(g^{\theta\dagger})$ is in a subspace defined as follows.

Proposition 1. *Given the best response follower $g^{\theta\dagger}$, let $\psi^{\theta\dagger}$ be the expected discounted features under the leader f_θ and its best response follower $g^{\theta\dagger}$ defined as*

$$\psi^{\theta\dagger}(s,b) = \mathbb{E}_{p^\theta, g^{\theta\dagger}}\left[\sum_{t=0}^{\infty} \gamma_F^t \phi(s_t, b_t) \,\middle|\, \begin{array}{l} s_0 = s \\ b_0 = b \end{array}\right].$$

Let $\Sigma^{\theta\dagger} = \mathbb{E}_{d_{\gamma_F}^{\theta, g^{\theta\dagger}}}\left[\mathrm{Cov}_{g^{\theta\dagger}}[\psi^{\theta\dagger}(s,b)]\right]$. Then, the solution to the IRL lies in the kernel of $\Sigma^{\theta\dagger}$ centered at the ground truth parameter $\omega^\star$, i.e., $\mathrm{IRL}_\theta(g^{\theta\dagger}) = \{\omega \in \mathbb{R}^d : \Sigma^{\theta\dagger}(\omega - \omega^\star) = 0\}$. [The proof is provided in Section A.1.]

Since the follower's transition probability depends on θ and changes at each outer iteration, we naturally obtain expert policies on different MDPs sharing the same reward, the setting in which Cao et al. (2021); Rolland et al. (2022) identify the reward up to a constant offset. Consequently, we obtain the convergence of the estimated reward parameter up to a constant offset of the reward.

Theorem 1. *Let $\{(\theta_k, g^{\theta_k\dagger})\}_{k=1}^K$ be a sequence of leader parameters and their corresponding best-response followers. For each $k \in [K]$, let $\omega_k \in \mathrm{IRL}_{\theta_k}(g^{\theta_k\dagger})$ be an optimal reward parameter, and define the expected feature covariance matrix as $\Sigma_k = \mathbb{E}_{d_{\gamma_F}^{\theta_k, g^{\theta_k\dagger}}}\left[\mathrm{Cov}_{g^{\theta_k\dagger}}[\psi^{\theta_k\dagger}(s,b)]\right]$, and let $\bar{\Sigma}_K = \sum_{k=1}^K \Sigma_k$. Then, any solution $\bar{\omega}_K$ to $\sum_{k=1}^K \Sigma_k \omega = \sum_{k=1}^K \Sigma_k \omega_k$ satisfies $\bar{\omega}_K \in \bigcap_{k=1}^K \mathrm{IRL}_{\theta_k}(g^{\theta_k\dagger})$. Furthermore, let $\Delta_{\mathrm{const}} \in \mathbb{R}^d$ be a vector satisfying $\langle \phi(s,b), \Delta_{\mathrm{const}} \rangle = 1$ for all $(s,b) \in \mathcal{S} \times \mathcal{B}$ (if such a vector exists, and 0 otherwise). If the aggregated covariance matrix satisfies the rank condition $\textsc{Rank}\left(\bar{\Sigma}_K\right) = d - \mathbb{I}\{\Delta_{\mathrm{const}} \neq 0\}$, then we have $\bar{\omega}_K = \omega^\star + c\Delta_{\mathrm{const}}$ for some $c \in \mathbb{R}$. Consequently, the recovered reward function satisfies $r_{\bar{\omega}_K}(s,b) = r_{\omega^\star}(s,b) + c$. [The proof is provided in Section A.2.]*

To solve the system $\sum_{k=1}^{K} \Sigma_k \omega = \sum_{k=1}^{K} \Sigma_k \omega_k$ reliably with finite samples, we standardize the covariance matrix so that the eigenvalues are either 1 or 0 by dropping eigenvalues smaller than a threshold ϵ, then solve $\sum_{k=1}^{K} U_k \omega = \sum_{k=1}^{K} U_k \omega_k$ with the standardized covariance matrix U, whose solution space is the same as the original system. The algorithmic detail is provided in Algorithm 2. The resulting reward parameter $\bar{\omega}_K$ is guaranteed to converge to the ground-truth reward parameter up to a shift in the direction of Δ_{const} under mild conditions, shown in Corollary 1.

The eigenvalue truncation plays three distinct roles, which we separate here to avoid conflating them. (i) *Support for bias cancellation.* With finitely many trajectories, the empirical covariance matrix is generically of full rank, and the directions that form the kernel of the exact Σ_k appear as small non-zero eigenvalues. Truncation discards these directions and restores the kernel characterized in Proposition 1. It does not alter the identifiability limit itself, which is dictated by Theorem 1. (ii) *Numerical stabilization.* In tasks whose feature distribution is heavily skewed, such as the regulator tasks considered here, truncation is what keeps the aggregated linear system solvable to an acceptable accuracy. (iii) *Implicit regularization.* By avoiding division by near-zero eigenvalues, truncation bounds the norm of the recovered reward, which in turn bounds the influence term and prevents the HG norm from exploding. Roles (i) and (ii) are what Algorithm 2 and Corollary 1 are designed for; role (iii) is an empirical effect that we observe consistently in Section 5, and we do not claim it as a theoretical guarantee.

Corollary 1. *Let $\{U_k\}_{k=1}^{K}$ be the sequence defined in Algorithm 2, let $\bar{U}_K = \sum_{k=1}^{K} U_k$, and let $\widehat{\omega}_k$ be the sequence of estimated reward vectors. Suppose that $\widehat{\omega}_k$ are independently distributed, each $\widehat{\omega}_k$ satisfies $\mathbb{E}[U_k \widehat{\omega}_k] = U_k \omega_k$ and $\mathrm{Cov}[U_k \widehat{\omega}_k] \preccurlyeq C \cdot I$ for some universal constant $C \geqslant 0$, and $\textsc{Rank}\left(\sum_{k=1}^{K} U_k\right) = d - \mathbb{I}\{\Delta_{\text{const}} \neq 0\}$ for all $K \geqslant \bar{K}$ for some $\bar{K} \geqslant 1$. Let $W = I - \|\Delta_{\text{const}}\|_2^{-2} \Delta_{\text{const}} \Delta_{\text{const}}^{\top}$ (which must be regarded as $W = I$ if $\Delta_{\text{const}} = 0$), and let $\Delta_{\omega}^{K} = W(\bar{\omega}_K - \omega^{\star})$. Then, the output of Algorithm 2 satisfies $\mathbb{E}[\Delta_{\omega}^{K}] = 0$ and $\mathrm{Cov}[\Delta_{\omega}^{K}] \preccurlyeq C \cdot K \cdot (\bar{U}_K^{-})^2$, where $\bar{U}_K^{-}$ is the Moore-Penrose pseudo-inverse of $\bar{U}_K$. In particular, if the greatest eigenvalue of $\bar{U}_K^{-}$ scales in $o(K^{-1/2})$, we have $\lim_{K \to \infty} \Delta_{\omega}^{K} = 0$ in probability, implying that the true reward is recovered up to a constant offset. [The proof is provided in Section A.3.]*

Corollary 1 is a conditional statement about the aggregation step: it treats the sequence $\{U_k\}$ as fixed in advance and assumes the estimates $\widehat{\omega}_k$ to be independent. Neither holds verbatim in Algorithm 1, where θ_k is selected adaptively and U_k and $\widehat{\omega}_k$ are computed from the same finite batch. Section A.4 states the assumptions under which the conclusion survives in the adaptive case, and makes explicit which variant of the algorithm the resulting guarantee covers.

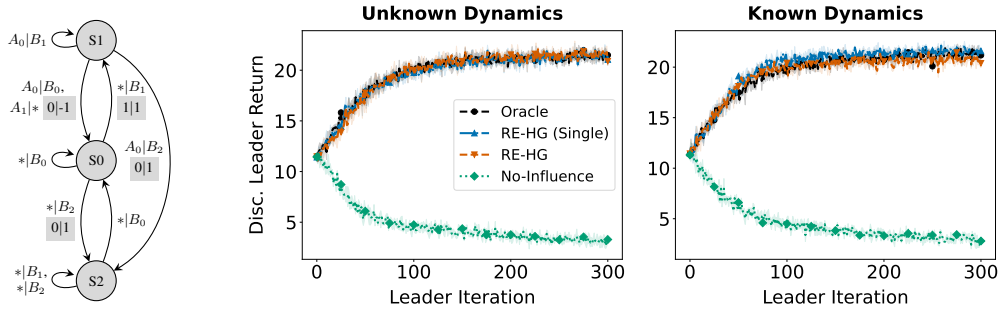
The ground-truth reward vector recovery translates to the true HG recovery.

Proposition 2. *Let $\omega^{\star}$ be the ground truth reward vector and $\widehat{\omega}$ be the estimated reward vector. Let $c = \langle \widehat{\omega} - \omega^{\star}, \Delta_{\text{const}} \rangle / \|\Delta_{\text{const}}\|_2^2$ and $\Delta_{\omega} = \widehat{\omega} - \omega^{\star} - c\Delta_{\text{const}}$. Then, the optimal soft-value functions $V^{\star}$ and $\widehat{V}$ corresponding to $\omega^{\star}$ and $\widehat{\omega}$ satisfy $\|\widehat{V} - V^{\star} - c(1-\gamma_F)^{-1}\|_{\infty} \leqslant L(1-\gamma_F)^{-1}\|\Delta_{\omega}\|_2$, where $L \geqslant \|\phi(s,b)\|_2$ for all $(s,b) \in \mathcal{S} \times \mathcal{B}$. Consequently, the error in the HG computation due to the error of the reward estimation is $O(\|\Delta_{\omega}\|_2)$. [The proof is provided in Section A.5.]*

Implementation Algorithm 1 shows the overall framework of the proposed approach, *Reward-Estimated Hypergradient RE-HG*. At each upper-level iteration k, the follower first learns its best response under θ_k. In our setting, the leader cannot intervene in this process. The trajectories, $\mathcal{D}_k$, are then collected for the following IRL steps (Lines 5–6). The HG is computed by using the trajectories, $\mathcal{R}_k$, obtained in simulation with the surrogate follower $\widehat{g}_k$ (Lines 9–16). For the implementation, we remark on two points. First, we use the trajectories $\mathcal{R}_k$ rather than $\mathcal{D}_k$ for HG estimation. As we explicitly utilize $\widehat{g}_k$ itself for the HG estimation, using $\mathcal{R}_k$ will relax the bias due to the inconsistent use of $\widehat{g}_k$ and $\mathcal{D}_k$. Second, while the overall framework does not change, environment-specific implementation, for example, the use of dynamic programming when the transition dynamics are known, can simplify and stabilize the resulting algorithm. The details are discussed in Section C and summarized in Table 2.

Algorithm 1 The RE-HG Algorithm

Initialize leader parameters θ^0 and leader's critic Q_L^0
for $k = 0$ to $N-1$ **do**
  (Follower) Learn the best response $g^{\theta_k\dagger}$ for $\mathcal{M}_{\theta_k}$
  Collect the trajectories $\mathcal{D}_k = \{\tau_k^n\}$, compute feature covariance Σ_k
  Perform IRL to obtain reward vector ω_k, surrogate policy $\widehat{g}_k$
  Update the aggregated reward vector $\bar{\omega}_k$ by Algorithm 2
  Re-derive surrogate policy $\widehat{g}_k$, action value function Q_F^k, and state value function V_F^k using $\bar{\omega}_k$
  Collect the trajectories $\mathcal{R}_k = \{\tau_k^n\}$ by simulating $\widehat{g}_k$ on $\mathcal{M}_{\theta_k}$
  $Q_L^{k+1} \leftarrow \texttt{CriticUpdate}(Q_L^k, \mathcal{R}_k)$ # by Algorithm 3 or Algorithm 4
  for $(\bar{s}, b) \in \tau \in \mathcal{R}_k$ (where $\bar{s} = s \in \mathcal{S}$ for Conf-MDPs, $\bar{s} = (s, a) \in \mathcal{S} \times \mathcal{A}$ for MGs) **do**
    $V_L^{k+1}(\bar{s}) \leftarrow \mathbb{E}_{b\sim\widehat{g}_k(\cdot|\bar{s})}\left[Q_L^{k+1}(\bar{s}, b)\right]$
    $B_L^{k+1}(\bar{s}, b) \leftarrow Q_L^{k+1}(\bar{s}, b) - V_L^{k+1}(\bar{s})$
    $\widehat{\nabla_{\theta_k} Q_F^k}(\bar{s}, b) \leftarrow \texttt{FollowerQGrad}(\bar{s}, b, \mathcal{R}_k, V_F^k)$ # by Algorithm 7
  $\widehat{\partial_{\theta_k} J_L^k} \leftarrow \texttt{PartialDerivative}(\mathcal{R}_k, \theta^k, V_L^{k+1}, Q_L^{k+1})$ # by Algorithm 5 or Algorithm 6
  $\widehat{\nabla_{\theta_k} J_L^k} \leftarrow \widehat{\partial_{\theta_k} J_L^k} + \text{InfluenceTerm}(B_L^{k+1}, \widehat{\nabla_{\theta_k} Q_F^k})$ # by Algorithm 8 or Algorithm 9
  $\theta^{k+1} \leftarrow \theta^k + \alpha \widehat{\nabla_{\theta_k} J_L^k}$



(a) Environment (b) Discounted leader return under unknown dynamics (left) and known dynamics (right)

Figure 1: Discrete toy task. (a) State transition diagram. $r_L \mid r_F$ represents the rewards. (b) Leader's discounted return across 3 independent training runs. For each run, performance is evaluated as the mean of 10 test rollouts. Avg. (line) and std. (shaded area) across the 3 runs are displayed.

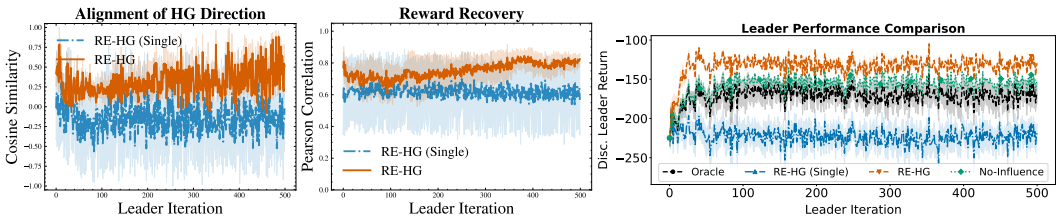
5 Experiments

We evaluate the robustness of RE-HG against an ORACLE baseline that computes HGs with full access to the ground-truth rewards and transition dynamics. This baseline relies on stronger assumptions than the standard white-box follower assumption utilized in Thoma et al. (2024); Kudo et al. (2026), as it requires access to the transition dynamics. As an idealized model, it represents the empirical upper bound and serves as our primary benchmark for gradient accuracy and overall performance.

2-Player Markov Games (Discrete Case) To assess the necessity of the influence term and the HG accuracy of RE-HG, we design an environment (Figure 1a) with an inherent leader-follower conflict.

The leader earns $r_L = +1$ only on $S_0 \to S_1$, favoring an $S_0 \leftrightarrow S_1$ loop. Conversely, the follower prefers $S_0 \to S_1 \to S_2$ to maximize $r_F = +1$. At S_1, the leader can choose A_1 to force the loop, but this imposes a penalty $r_F = -1$. If the leader greedily persists with A_1, the follower will adapt by transitioning $S_0 \to S_2$ to avoid the penalty, eventually depriving the leader of all rewards. Thus, the leader must strategically guide the follower to S_1 while preventing deflection to S_2 by choosing A_0 at appropriate intervals.

As observed in Figure 1b, RE-HG and ORACLE significantly outperform NO-INFLUENCE, a variant ignoring the influence term. It clearly indicates the need to estimate the influence term. Specifically, the leader policies in RE-HG and ORACLE converge to $f(A_0|S_1) \approx 1.0$. In contrast, NO-INFLUENCE leader policy converges to $f(A_0|S_1) \approx 0.0$, resulting in the breakdown described



(a) HG Similarity (b) Reward Correlation (c) Leader's Discounted Return

Figure 2: Results on Building Thermal Control Task. (a) Cosine similarity of the HG between the estimated and Oracle models. (b) Pearson correlation between the recovered and true rewards, weighted by the undiscounted state-action occupancy measure of the surrogate follower policy. (c) Leader's discounted return. For each run, performance is evaluated as the mean of 50 test rollouts. Avg. (line) and std. (shaded area) across the 5 independent training runs are displayed.

above. The relatively low Pearson correlation stems from estimation errors in unvisited regions, which do not affect the gradient direction of the HG. In fact, RE-HG maintains a high cosine similarity. Essentially, RE-HG successfully recovers the global reward structure necessary for optimization. Further analysis of the Pearson and cosine similarities and experimental details are provided in Section C.3.

Configurable MDP (Continuous Case) We evaluate the effectiveness of the aggregation mechanism in RE-HG for mitigating estimation instability by comparing it against RE-HG(SINGLE), a variant performing IRL only once at the beginning. The task is under-actuated temperature regulation in a building with $n = 8$ zones. The state $s_t \in \mathbb{R}^n$ is the temperature deviation, the follower's action $b \in \mathbb{R}^m$ ($m = 4$) is the HVAC control, and the leader configures the insulation and airflow $\theta = (\alpha, \nu) \in [0, 1]^{2n}$. The system follows linear-Gaussian dynamics $s' = A_\theta s + Bb + w$, where $w \sim \mathcal{N}(\mathbf{0}, \Sigma_w)$ represents unobservable disturbances. The follower maximizes an entropy-regularized quadratic reward $r_F(s, b) = -s^\top W_F s - b^\top U_F b$, while the leader minimizes insulation costs and thermal instability. Crucially, because the most costly zone lacks direct HVAC control, the follower must regulate it indirectly via heat transfer influenced by the leader's configuration. In this experiment, identical gradient norm clipping with a maximum norm of 1.0 is applied to every method.

Comparisons in Figure 2c confirm that the proposed method, RE-HG, overcomes the estimation instability of RE-HG(SINGLE) and clearly outperforms it. Lacking bias cancellation and eigenvalue truncation, RE-HG(SINGLE) recovers the excessively steep reward, causing the gradient norm to reach the order of 10^7. Because estimation errors propagate directly into the gradient, the optimization is misled into an inappropriate local minimum, resulting in significant performance degradation. In contrast, RE-HG suppresses the reward norm through the aforementioned two implicit regularization effects of the aggregation process, keeping the gradient magnitude around 10^1. Although the values are not perfectly identical due to the scale suppression effect, Figures 2a and 2b show that RE-HG achieves a higher cosine similarity with ORACLE and a higher Pearson correlation with the true reward weighted by the surrogate follower policy compared to RE-HG(SINGLE). This indicates that the global gradient direction and reward structure are appropriately recovered.

As a secondary effect, this smoothing leads RE-HG to attain a higher return than ORACLE. We stress that this is an optimization effect and not an indication that RE-HG estimates the HG more accurately. While ORACLE computes the exact true gradient, the follower's skewed cost matrices (W_F, U_F) create a steep reward landscape, causing the gradient scale to explode (around 10^3). Consequently, terms other than the influence term cannot be properly accounted for, severely distorting the overall gradient direction and trapping ORACLE in a suboptimal local minimum (behavior similar to NO-INFLUENCE converging to a local minimum). On the other hand, by recovering a smoothed reward bounded within a moderate scale (0.2 to 3.2) rather than the steep true reward, RE-HG obtains a gradient landscape on which all terms are appropriately balanced, avoiding the sharp local optima that trap ORACLE. This gap is not explained by the magnitude of the update alone: Section C.6 reports ORACLE variants with a learning rate, gradient clipping, and a scaled influence term, none of which reaches the performance of RE-HG.

Further details, including the improvement of cosine similarity and Pearson correlation when the steep reward setting is relaxed, and experimental setups, are provided in Section C.4.

2-Player Markov Games (Continuous Case) We evaluate the HG accuracy of RE-HG on a continuous MG formulated as a Linear Quadratic Regulator (LQR) task. The dynamics are $s_{t+1} = As_t + Bb_t + Ca_t$, where $s_t \in \mathbb{R}^2$ is the state (starting at $(1, 0)$), $b_t \in \mathbb{R}^1$ is the follower's control, and $a_t \in \mathbb{R}^1$ is the leader's interference. Both agents employ Gaussian policies. The follower aims to stabilize at $s_F^\star = (0, 0)$ by maximizing a quadratic reward $r_F(s, b) = -s^\top W_F s - b^\top U_F b$, where counteracting the leader's interference increases control costs. Conversely, the leader seeks to induce oscillation around $s_L^\star = (-0.8, 0)$ by maximizing $r_L(s, a, b) := \exp(-\frac{1}{2}(s - s_L^\star)^\top M(s - s_L^\star))$. This explicit conflict of interest tests whether the leader can strategically guide the follower by accounting for its cost structure via HG, rather than relying on inefficient brute-force interference.

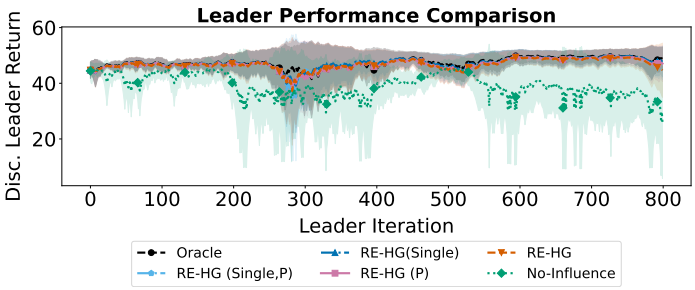


Figure 3: Results on 2-player MG, averaged over 3 independent training runs (10 test rollouts per run). Avg. (line) and std. (shaded area) are displayed. "P" indicates the known transition probability scenario.

Figure 3 shows that RE-HG achieves comparable performance to ORACLE, effectively navigating the true gradient direction despite the presence of estimation errors. Furthermore, it is more stable than NO-INFLUENCE. Both the HG cosine similarity relative to ORACLE and the weighted Pearson correlation consistently remained near 1.0 across the majority of the iterations for RE-HG. Figures illustrating these metrics, along with full experimental settings, are provided in Section C.5.

6 Conclusion

We introduced RE-HG, an algorithm for optimizing the leader's parameters via HG in a bilevel RL setting with a black-box follower. Our theoretical analysis revealed that HG estimation is fundamentally vulnerable to reward shaping, an unavoidable issue in single-environment IRL. To resolve this, we established a bias cancellation theory showing that, under a linear reward assumption, the here-proposed aggregated IRL mathematically neutralizes errors caused by potential-based shifts. Based on this theory, we developed a reward recovery algorithm enhanced by a norm suppression smoothing effect. Empirically, despite operating without the restrictive white-box follower assumption, RE-HG achieved performance competitive with a white-box Oracle baseline by yielding accurate gradient direction estimates.

Limitations RE-HG has several limitations. First, our theoretical framework relies on the assumption of an optimal follower. Whether RE-HG operates stably when the follower follows sub-optimal learning trajectories remains open for discussion; Section C.9 reports an empirical study under two types of suboptimal followers, but our analysis does not cover this case. Second, formulating the IRL as an inner-loop min-max optimization inherently introduces estimation variance that directly propagates to the outer-loop HG. Stabilizing the influence term remains an open challenge. Third, our bias cancellation theory is built upon the linear reward assumption with a known feature extractor. Robustness to a misspecified feature basis is examined empirically in Section C.8, but extending the theory to non-linear reward structures requires further theoretical and empirical investigation; integrating multi-environment IRL techniques designed for non-linear representations is a natural direction. Finally, RE-HG removes the need for the follower's reward function, but not the need for environment access: computing and rolling out the surrogate best response requires either the transition model or a simulator. Integrating offline IRL is a natural direction for removing this dependence.

Acknowledgments and Disclosure of Funding

This work was partially supported by JSPS KAKENHI Grant Number 23H00483 and 26K02993.

References

Albrecht, S., Leibold, M., and Ulbrich, M. (2012). A bilevel optimization approach to obtain optimal cost functions for human arm movements. *Numerical Algebra, Control and Optimization*, 2(1):105–127.

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., van Hasselt, H., and Silver, D. (2017). Successor features for transfer in reinforcement learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4058–4068, Red Hook, NY, USA. Curran Associates Inc.

Cao, H., Cohen, S. N., and Szpruch, L. (2021). Identifiability in inverse reinforcement learning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA. Curran Associates Inc.

Chakraborty, S., Bedi, A. S., Koppel, A., Wang, H., Manocha, D., Wang, M., and Huang, F. (2024). PARL: A unified framework for policy alignment in reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Fu, J., Luo, K., and Levine, S. (2018). Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations*.

Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., Xu, R., Zhang, R., Ma, S., Bi, X., et al. (2025). Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. (2017). Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1352–1361. JMLR.org.

Ho, J. and Ermon, S. (2016). Generative adversarial imitation learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 4572–4580, Red Hook, NY, USA. Curran Associates Inc.

Hong, M., Wai, H., Wang, Z., and Yang, Z. (2023). A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM J. Optim.*, 33(1):147–180.

Kudo, M., Tanabe, T., Wachi, A., and Akimoto, Y. (2026). Sample-efficient hypergradient estimation for decentralized bi-level reinforcement learning. In *ICAPS '26: Proceedings of the Thirty-Sixth International Conference on Automated Planning and Scheduling*.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.*, 17(1):1334–1373.

Ng, A. Y., Harada, D., and Russell, S. J. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, page 278–287, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Ni, T., Sikchi, H., Wang, Y., Gupta, T., Lee, L., and Eysenbach, B. (2020). f-irl: Inverse reinforcement learning via state marginal matching. In *Conference on Robot Learning*.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Rolland, P., Viano, L., Schürhoff, N., Nikolov, B., and Cevher, V. (2022). Identifiability and generalizability from multiple experts in inverse reinforcement learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Samvelyan, M., Rashid, T., Schroeder de Witt, C., Farquhar, G., Nardelli, N., Rudner, T. G. J., Hung, C.-M., Torr, P. H. S., Foerster, J., and Whiteson, S. (2019). The starcraft multi-agent challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, page 2186–2188, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Shen, H., Yang, Z., and Chen, T. (2024). Principled penalty-based methods for bilevel reinforcement learning and rlhf. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Shi, B., Zhai, L., and Liu, C. (2025). Stackelberg game based on trajectory prediction for lane change in mixed traffic. *IEEE Access*, 13:135196–135207.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T. P., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nat.*, 529(7587):484–489.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning, Second Edition: An Introduction*. MIT Press.

Syed, U., Bowling, M., and Schapire, R. E. (2008). Apprenticeship learning using linear programming. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1032–1039, New York, NY, USA. Association for Computing Machinery.

Thoma, V., Pásztor, B., Krause, A., Ramponi, G., and Hu, Y. (2024). Contextual bilevel reinforcement learning for incentive alignment. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C., editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Yang, Y., Gao, B., and Yuan, Y. (2025). Bilevel reinforcement learning via the development of hyper-gradient without lower-level convexity. In Li, Y., Mandt, S., Agrawal, S., and Khan, M. E., editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2025, Mai Khao, Thailand, 3-5 May 2025*, Proceedings of Machine Learning Research, pages 4780–4788. PMLR.

Zeng, S., Li, C., Garcia, A., and Hong, M. (2022). Maximum-likelihood inverse reinforcement learning with finite-time guarantees. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 10122–10135. Curran Associates, Inc.

Zhou, Z., Bloem, M., and Bambos, N. (2018). Infinite time horizon maximum causal entropy inverse reinforcement learning. *IEEE Transactions on Automatic Control*, 63(9):2787–2802.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the paper's three main contributions. These claims are supported by the methodology and experimental evidence.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We include a dedicated "Limitations" subsection in conclusion. This section discusses the assumption.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: Necessary assumptions, such as Reward Linearity, are explicitly defined in the text. Complete formal proofs and Propositions are provided in Appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

[Yes]

Justification: The paper provides comprehensive pseudo-code, exhaustive hyperparameter lists, and detailed implementation settings. Furthermore, the number of independent trials for statistical significance is explicitly stated in all figure captions.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: The anonymized source code and instructions for reproducing the results are included in the supplementary material.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Comprehensive details of the experimental settings are provided in the appendix. Theoretical formulations and algorithms are also clearly documented to ensure reproducibility.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: All experimental results in the paper and appendix report the mean values and standard deviations across multiple independent runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Detailed specifications of the computing resources, including the CPU model, memory capacity, and total wall-clock time for the experiments, are provided in Section E.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We confirm that our research, which focuses on foundational reinforcement learning algorithms, fully conforms to these guidelines.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss the broader impact in Section D.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This research focuses on fundamental reinforcement learning algorithms and does not involve models or datasets that pose a high risk of misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [N/A]

    Justification: This research does not use any existing research-specific assets, such as third-party datasets, pretrained models, or published code packages.

    Guidelines:

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

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The source code for our proposed algorithm is included in the supplementary material, along with a README file providing instructions for installation and reproduction.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This research does not involve crowdsourcing or research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This research does not involve human subjects, making IRB approval inapplicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were used strictly for language editing and proofreading of the manuscript, and were not involved in the development of the core methodology or experimental design.

    Guidelines:

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

A Proofs

A.1 Proposition 1

Proof. As is claimed in Zeng et al. (2022), the process of IRL_θ can be interpreted as a maximum likelihood estimation of $g^{\theta\dagger}$ by the policy $g_r^\star$ optimal for the reward model r, where the reward parameter is considered as the parameter of the policy itself. Indeed, we observe

$$\begin{aligned}
\mathrm{IRL}_\theta(g^{\theta\dagger}) &= \operatorname*{argmin}_r \max_g J_\theta(g;r) - \mathbb{E}_{\rho_0^\theta, p^\theta, g^{\theta\dagger}}\left[\sum_{t=0}^\infty \gamma^t r(s_t,b_t)\right] \\
&= \operatorname*{argmin}_r \max_g \mathbb{E}_{\rho_0^\theta}\left[V_{r,g}^{\mathrm{soft}}(s_0)\right] - \mathbb{E}_{\rho_0^\theta, p^\theta, g^{\theta\dagger}}\left[\sum_{t=0}^\infty \gamma^t r(s_t,b_t)\right] \\
&= \operatorname*{argmin}_r \mathbb{E}_{\rho_0^\theta}\left[V_{r,g_r^\star}^{\mathrm{soft}}(s_0)\right] - \mathbb{E}_{\rho_0^\theta, p^\theta, g^{\theta\dagger}}\left[\sum_{t=0}^\infty \gamma^t r(s_t,b_t)\right] \\
&= \operatorname*{argmin}_r \mathbb{E}_{\rho_0^\theta, p^\theta, g^{\theta\dagger}}\left[\sum_{t=0}^\infty \gamma^t \left(V_{r,g_r^\star}^{\mathrm{soft}}(s_t) - \gamma V_{r,g_r^\star}^{\mathrm{soft}}(s_{t+1})\right)\right] - \mathbb{E}_{\rho_0^\theta, p^\theta, g^{\theta\dagger}}\left[\sum_{t=0}^\infty \gamma^t r(s_t,b_t)\right] \\
&= \operatorname*{argmax}_r \mathbb{E}_{\rho_0^\theta, p^\theta, g^{\theta\dagger}}\left[\sum_{t=0}^\infty \gamma^t \left(r(s_t,b_t) + \gamma V_{r,g_r^\star}^{\mathrm{soft}}(s_{t+1}) - V_{r,g_r^\star}^{\mathrm{soft}}(s_t)\right)\right] \\
&= \operatorname*{argmax}_r \mathbb{E}_{\rho_0^\theta, p^\theta, g^{\theta\dagger}}\left[\sum_{t=0}^\infty \gamma^t \left(Q_{r,g_r^\star}^{\mathrm{soft}}(s_t,b_t) - V_{r,g_r^\star}^{\mathrm{soft}}(s_t)\right)\right] \\
&= \operatorname*{argmax}_r \mathbb{E}_{d_{\gamma_F}^{\theta, g^{\theta\dagger}}}\left[\mathbb{E}_{g^{\theta\dagger}}\left[\log g_r^\star(s,b)\right]\right].
\end{aligned}$$

Let $\ell(s,b;\omega) = \log g_{r_\omega}^\star(s,b)$. Let V_ω^{soft} and Q_ω^{soft} be the optimal soft value functions under the reward model r_ω. Then, we have

$$\begin{aligned}
\nabla_\omega \ell(s,b;\omega) &= \nabla_\omega Q_\omega^{\mathrm{soft}}(s,b) - \nabla_\omega V_\omega^{\mathrm{soft}}(s) \\
&= \nabla_\omega Q_\omega^{\mathrm{soft}}(s,b) - \mathbb{E}_{g_\omega^\star}[\nabla_\omega Q_\omega^{\mathrm{soft}}(s,b)],
\end{aligned}$$

where we used the soft-Bellman equation for value function, $V_\omega^{\mathrm{soft}}(s) = \log \int \exp(Q_\omega^{\mathrm{soft}}(s,b))\mathrm{d}b$. On the other hand, using the soft-Bellman equation for action value function, we obtain

$$\nabla_\omega Q_\omega^{\mathrm{soft}}(s,b) = \phi(s,b) + \gamma \mathbb{E}_{s'}\left[\mathbb{E}_{g_\omega^\star}[\nabla_\omega Q_\omega^{\mathrm{soft}}(s',b')]\right].$$

The solution to this recursive formula is equivalent to the expected discounted features, i.e., $\nabla_\omega Q_\omega^{\mathrm{soft}}(s,b) = \psi_\omega(s,b)$. Therefore, we have

$$\nabla_\omega \ell(s,b;\omega) = \psi_\omega(s,b) - \mathbb{E}_{g_\omega^\star}[\psi_\omega(s,b)].$$

Moreover, the Hessian is

$$\begin{aligned}
\nabla_\omega^2 \ell(s,b;\omega) &= \nabla_\omega \psi_\omega(s,b) - \mathbb{E}_{b'\sim g_\omega^\star}[\psi_\omega(s,b')\nabla \ell(s,b';\omega)^\top] - \mathbb{E}_{b'\sim g_\omega^\star}[\nabla_\omega \psi_\omega(s,b')] \\
&= \nabla_\omega \psi_\omega(s,b) - \mathbb{E}_{b'\sim g_\omega^\star}[\nabla_\omega \psi_\omega(s,b')] - \mathrm{Cov}_{b'\sim g_\omega^\star}[\psi_\omega(s,b')].
\end{aligned}$$

Let $L(\omega) = \mathbb{E}_{d_{\gamma_F}^{\theta, g^{\theta\dagger}}}\left[\mathbb{E}_{g^{\theta\dagger}}\left[-\ell(s,a;\omega)\right]\right]$. Noting that $g^{\theta\dagger} = g_{\omega^\star}^\star$, the Hessian of this loss function at $\omega = \omega^\star$ is

$$\begin{aligned}
\nabla_\omega^2 L(\omega^\star) &= -\mathbb{E}_{d_{\gamma_F}^{\theta, g^{\theta\dagger}}}\left[\mathbb{E}_{g^{\theta\dagger}}\left[\nabla_\omega^2 \ell(s,b;\omega^\star)\right]\right] \\
&= -\mathbb{E}_{d_{\gamma_F}^{\theta, g^{\theta\dagger}}}\left[\mathbb{E}_{g^{\theta\dagger}}\left[\nabla_\omega \psi_{\omega^\star}(s,b) - \mathbb{E}_{g_{\omega^\star}^\star}[\nabla_\omega \psi_{\omega^\star}(s,b)] - \mathrm{Cov}_{g_{\omega^\star}^\star}[\psi_{\omega^\star}(s,b)]\right]\right] \\
&= -\mathbb{E}_{d_{\gamma_F}^{\theta, g_{\omega^\star}}}\left[\mathbb{E}_{g_{\omega^\star}}\left[\nabla_\omega \psi_{\omega^\star}(s,b)\right] - \mathbb{E}_{g_{\omega^\star}^\star}[\nabla_\omega \psi_{\omega^\star}(s,b)] - \mathrm{Cov}_{g_{\omega^\star}^\star}[\psi_{\omega^\star}(s,b)]\right] \\
&= \mathbb{E}_{d_{\gamma_F}^{\theta, g_{\omega^\star}^\star}}\left[\mathrm{Cov}_{g_{\omega^\star}^\star}[\psi_{\omega^\star}(s,b)]\right].
\end{aligned}$$

Hence, it is non-negative definite. If two reward functions r_0 and r_1 admit the same optimal policy, then their convex combination $r_\lambda = \lambda r_0 + (1-\lambda) r_1$ for $\lambda \in [0,1]$ also admits the same optimal policy. This implies that all the optimal reward parameter must lie in the kernel of $\nabla^2_\omega L(\omega^\star)$ centered at $\omega^\star$. This completes the proof. □

A.2 Theorem 1

Proof. From Proposition 1, we have $\Sigma_k(\omega_k - \omega_\star) = 0$ for each $k \in [K]$. Then, we have

$$\bar{\Sigma}_K \bar{\omega}_K = \sum_{k=1}^K \Sigma_k \omega_k = \sum_{k=1}^K \Sigma_k \omega^\star = \bar{\Sigma}_K \omega^\star.$$

Therefore, we have $\bar{\Sigma}_K(\bar{\omega}_K - \omega^\star) = 0$, which indicates that $\bar{\omega} - \omega^\star$ lies in the kernel of $\bar{\Sigma}_K$. Because Σ_k are all non-negative definite, the kernel of $\bar{\Sigma}_K$ is the intersection of the kernels of Σ_k. Hence, we have $\bar{\omega}_K \in \mathrm{IRL}_{\theta_k}(g^{\theta_k \dagger})$ for all $k \in [K]$.

The rest is straightforward. If $\Delta_{\mathrm{const}} = 0$ and $\textsc{Rank}(\bar{\Sigma}_K) = d$, there exists a unique solution to $\bar{\Sigma}_K \omega = \sum_{k=1}^K \Sigma_k \omega_k$ and it must agree with $\omega^\star$. If $\Delta_{\mathrm{const}} \neq 0$ and $\textsc{Rank}(\bar{\Sigma}_K) = d-1$, then Δ_{const} must lie in the kernel of $\bar{\Sigma}_K$ because $\omega + c\Delta_{\mathrm{const}}$ results in a constant offset of the reward function as

$$r_{\omega + c\Delta_{\mathrm{const}}}(s,b) = \langle \omega + c\Delta_{\mathrm{const}}, \phi(s,b) \rangle = \langle \omega, \phi(s,b) \rangle + c,$$

and hence the loss function L must be constant in the direction of Δ_{const}. That is, the kernel of $\bar{\Sigma}_K$ is spanned by Δ_{const}. Therefore, we have $\bar{\omega}_K = \omega^\star + c\Delta_{\mathrm{const}}$. This completes the proof. □

A.3 Corollary 1

Proof. In this proof, we consider $K \geqslant \bar{K}$. Then, from the assumption $\textsc{Rank}(\bar{U}_K) = d - \mathbb{I}\{\Delta_{\mathrm{const}} \neq 0\}$, the kernel of $\bar{U}_K$ must admit the kernel of W. Therefore, $W = \bar{U}_K \bar{U}_K^- = \bar{U}_K^- \bar{U}_K$. Moreover, $W\bar{U}_K^- = \bar{U}_K^-$ and $W\bar{U}_K = \bar{U}_K$ because of the property of pseudo-inverse.

Note first that $\bar{\omega}_K$ is a solution to $\bar{U}_K \omega = \sum_{k=1}^K U_k \widehat{\omega}_k$. The solution can be represented as

$$\bar{\omega}_K = \sum_{k=1}^K U_k \widehat{\omega}_k + (I - W)\delta$$

for some $\delta \in \mathbb{R}^d$. Because $W = I - \|\Delta_{\mathrm{const}}\|_2^{-2} \Delta_{\mathrm{const}} \Delta_{\mathrm{const}}^\top$, we have

$$\bar{\omega}_K = \sum_{k=1}^K U_k \widehat{\omega}_k + \|\Delta_{\mathrm{const}}\|_2^{-2} \langle \delta, \Delta_{\mathrm{const}} \rangle \Delta_{\mathrm{const}}.$$

Letting $c = \|\Delta_{\mathrm{const}}\|_2^{-2} \langle \delta, \Delta_{\mathrm{const}} \rangle$, we have RE$\bar{\omega}_K = \sum_{k=1}^K U_k \widehat{\omega}_k + c\Delta_{\mathrm{const}}$. It indicates that

$$\begin{aligned}
\Delta_\omega^K &= W(\bar{\omega}_K - \omega^\star) \\
&= W\bar{U}_K^- \sum_{k=1}^K U_k \widehat{\omega}_k - W\omega^\star \\
&= \bar{U}_K^- \sum_{k=1}^K U_k \widehat{\omega}_k - \bar{U}_K^- \bar{U}_K \omega^\star \\
&= \bar{U}_K^- \sum_{k=1}^K U_k (\widehat{\omega}_k - \omega^\star).
\end{aligned}$$

From Proposition 1, we have $\Sigma_k(\omega_k - \omega^\star) = 0$, which is equivalent to $U_k(\omega_k - \omega^\star) = 0$ because Σ_k and U_k admit the same kernel by construction. Therefore, $\mathbb{E}[U_k \widehat{\omega}_k] = U_k \omega_k$ implies $\mathbb{E}[U_k \widehat{\omega}_k] =$

$U_k\omega^\star$. Then, we have

$$\begin{aligned}\mathbb{E}[\Delta_\omega^K] &= \bar{U}_K^- \sum_{k=1}^K U_k(\widehat{\omega}_k - \omega^\star) \\ &= \bar{U}_K^- \sum_{k=1}^K \mathbb{E}[U_k\widehat{\omega}_k] - U_k\omega^\star \\ &= 0.\end{aligned}$$

Because $\widehat{\omega}_k$ are independently distributed, we have

$$\begin{aligned}W\,\mathrm{Cov}[\bar{\omega}_K]W &= \mathbb{E}[\Delta_\omega^K(\Delta_\omega^K)^\top] \\ &= \bar{U}_K^- \sum_{k=1}^K \mathbb{E}[U_k(\widehat{\omega}_k - \omega^\star)(\widehat{\omega}_k - \omega^\star)^\top U_k]\bar{U}_K^- \\ &\preccurlyeq C \cdot K(\bar{U}_K^-)^2.\end{aligned}$$

In light of the multivariate Chebyschev's inequality, we have

$$\Pr[\|\Delta_\omega^K\|_{\mathrm{Cov}[\Delta_\omega^K]^-} > t] \leqslant \frac{d}{t^2}$$

for any $t > 0$. Let $\sigma_{\max}(\bar{U}_K^-)$ be the greatest eigenvalue of $\bar{U}_K^-$. Then, we have $\sigma_{\min}(\mathrm{Cov}[\Delta_\omega^K]^-) \geqslant (CK\sigma_{\max}(\bar{U}_K^-)^2)^{-1}$, where $\sigma_{\min}$ represents the smallest non-zero eigenvalue. Then, we have $\|\Delta_\omega^K\|_{\mathrm{Cov}[\Delta_\omega^K]^-} \geqslant \sqrt{\sigma_{\min}(\mathrm{Cov}[\Delta_\omega^K]^-)}\|\Delta_\omega^K\|_2$, and hence,

$$\Pr[\|\Delta_\omega^K\|_2 > t] \leqslant \frac{d}{t^2\sigma_{\min}(\mathrm{Cov}[\Delta_\omega^K]^-)} \leqslant \frac{dCK\sigma_{\max}(\bar{U}_K^-)^2}{t^2} \xrightarrow{K\to\infty} 0.$$

It implies $\lim_{K\to\infty}\Delta_\omega^K = 0$ in probability. This completes the proof. □

A.4 Adaptive Variant of Corollary 1

Σ_k is estimated from the batch $\mathcal{D}_k$ collected under θ_k, so U_k is not $\mathcal{F}_{k-1}$-measurable, and U_k and $\widehat{\omega}_k$, being functions of the same $\mathcal{D}_k$, are dependent.

Corollary 1 treats $\{U_k\}_{k=1}^K$ as a deterministic sequence and assumes the estimates $\widehat{\omega}_k$ to be independent. Neither holds in Algorithm 1: Σ_k is estimated from the batch $\mathcal{D}_k$ collected under θ_k, so U_k is not $\mathcal{F}_{k-1}$-measurable, and U_k and $\widehat{\omega}_k$, being functions of the same $\mathcal{D}_k$, are dependent.

We state below the assumptions under which the conclusion of Corollary 1 survives, and note in Remark 2 that one of them holds for a lagged or sample-split version of Algorithm 1 rather than for the algorithm as implemented.

Corollary 2. *Let $\{U_k\}_{k=1}^K$ and $\{\widehat{\omega}_k\}_{k=1}^K$ be the projection matrices and the estimated reward vectors generated by the closed loop of Algorithm 1, let $\bar{U}_K = \sum_{k=1}^K U_k$, and let $\mathcal{F}_k$ be the σ-algebra generated by the history $(\theta_1, \mathcal{D}_1, \widehat{\omega}_1, \ldots, \theta_k, \mathcal{D}_k, \widehat{\omega}_k)$, where $\mathcal{D}_k$ denotes the trajectories collected under θ_k. Let $b \geqslant 0$ and $C \geqslant 0$ be constants and suppose that, for every k,*

(i) U_k is $\mathcal{F}_{k-1}$-measurable;

(ii) $\mathbb{E}[U_k\widehat{\omega}_k \mid \mathcal{F}_{k-1}] = U_k\omega_k + U_k b_k$ for some $b_k \in \mathbb{R}^d$ with $\|b_k\|_2 \leqslant b$;

(iii) $\mathrm{Cov}[U_k\widehat{\omega}_k \mid \mathcal{F}_{k-1}] \preccurlyeq C \cdot I$;

and that

(iv) $\textsc{Rank}(\bar{U}_K) = d - \mathbb{I}\{\Delta_{\mathrm{const}} \neq 0\}$ for all $K \geqslant \bar{K}$, for some $\bar{K} \geqslant 1$.

Let W and Δ_ω^K be as in Corollary 1, let $\eta_k = U_k\widehat{\omega}_k - \mathbb{E}[U_k\widehat{\omega}_k \mid \mathcal{F}_{k-1}]$, and let $\sigma_{\min}$ denote the smallest non-zero eigenvalue. Then, for every $K \geqslant \bar{K}$, the output of Algorithm 2 decomposes as

$\Delta_\omega^K = \Delta_\omega^{K,\text{bias}} + \Delta_\omega^{K,\text{noise}}$ *with* $\Delta_\omega^{K,\text{bias}} = \bar{U}_K^- \sum_{k=1}^K U_k b_k$ *and* $\Delta_\omega^{K,\text{noise}} = \bar{U}_K^- \sum_{k=1}^K \eta_k$*, and these satisfy*

$$\mathbb{E}\left[\|\textstyle\sum_{k=1}^K \eta_k\|_2^2\right] \leqslant CdK, \tag{7}$$

$$\|\Delta_\omega^{K,\text{bias}}\|_2 \leqslant \frac{Kb}{\sigma_{\min}(\bar{U}_K)}. \tag{8}$$

Consequently, if $\Pr[\sigma_{\min}(\bar{U}_K) \geqslant \sqrt{K} a_K] \to 1$ *for some deterministic sequence* $a_K \to \infty$*, then* $\Delta_\omega^{K,\text{noise}} \to 0$ *in probability; and if* $\Pr[\sigma_{\min}(\bar{U}_K) \geqslant \alpha K] \to 1$ *for some* $\alpha > 0$*, then* $\Pr[\|\Delta_\omega^{K,\text{bias}}\|_2 > b/\alpha] \to 0$*. In particular, if* $b = 0$ *and the first spectral condition holds, then* $\Delta_\omega^K \to 0$ *in probability, which is the conclusion of Corollary 1 without the independence assumption.*

Proof. Throughout we take $K \geqslant \bar{K}$. Line 3 of Algorithm 2 sets $U_k = E I_{\Lambda \succ \epsilon} E^\top$ with E orthogonal and $I_{\Lambda \succ \epsilon}$ diagonal with entries in $\{0, 1\}$, so each U_k is an orthogonal projection; we use this in Steps 2 and 4.

Step 1. By (i), U_k is $\mathcal{F}_{k-1}$-measurable, so $\mathbb{E}[U_k \widehat{\omega}_k \mid \mathcal{F}_{k-1}] = U_k \mathbb{E}[\widehat{\omega}_k \mid \mathcal{F}_{k-1}]$ and hence

$$\eta_k = U_k \left(\widehat{\omega}_k - \mathbb{E}[\widehat{\omega}_k \mid \mathcal{F}_{k-1}]\right) \quad \text{and} \quad \mathbb{E}[\eta_k \mid \mathcal{F}_{k-1}] = 0. \tag{9}$$

Let $\xi_k = U_k \widehat{\omega}_k - U_k \omega_k$. By (ii), we have $\xi_k = U_k b_k + \eta_k$. Proposition 1 gives $\Sigma_k(\omega_k - \omega^\star) = 0$, and Σ_k and U_k have the same kernel by construction. Therefore, we have $U_k \omega_k = U_k \omega^\star$, and hence $\xi_k = U_k(\widehat{\omega}_k - \omega^\star)$. The derivation in Section A.3, which uses only (iv), applies verbatim and yields $\Delta_\omega^K = \bar{U}_K^- M_K$ with $M_K = \sum_{k=1}^K \xi_k$. Because $\xi_k = U_k b_k + \eta_k$, we have

$$\Delta_\omega^K = \bar{U}_K^- M_K = \bar{U}_K^- \sum_{k=1}^K \xi_k = \sum_{k=1}^K \bar{U}_K^- U_k b_k + \bar{U}_K^- \eta_k.$$

Step 2. Since U_k is an orthogonal projection, they are all symmetric and positive semi-definite. Therefore, so is $\bar{U}_K = \sum_{k=1}^K U_k$, and we have $\textsc{Range}(U_k) \subseteq \textsc{Range}(\bar{U}_K)$. It indicates that $U_k x \in \textsc{Range}(\bar{U}_K)$ holds for any $x \in \mathbb{R}^d$. Equivalently, for $x \in \textsc{Range}(\bar{U}_K)$, we have

$$\|\bar{U}_K^- x\|_2 \leqslant \frac{\|x\|_2}{\sigma_{\min}(\bar{U}_K)}. \tag{10}$$

Step 3. The sequence $(\eta_k)_k$ defined in (9) is by definition a Martingale difference sequence under the filtration $(\mathcal{F}_k)_k$. That is, by the tower property, η_k and η_j $(j < k)$ are uncorrelated. Then,

$$\begin{aligned}
\mathbb{E}\left[\left\|\sum_{k=1}^K \eta_k\right\|_2^2\right] &= \mathbb{E}\left[\sum_{k=1}^K \sum_{j=1}^K \eta_j^\top \eta_k\right] = \sum_{k=1}^K \sum_{j=1}^K \mathbb{E}\left[\eta_j^\top \eta_k\right] = \sum_{k=1}^K \mathbb{E}\left[\eta_k^\top \eta_k\right] \\
&= \sum_{k=1}^K \mathbb{E}\left[\operatorname{Tr}\left(\operatorname{Cov}[U_k \widehat{\omega}_k \mid \mathcal{F}_{k-1}]\right)\right] \leqslant CdK,
\end{aligned}$$

which proves (7).

Step 4. Since U_k is an orthogonal projection, Condition (ii) gives $\|U_k b_k\|_2 \leqslant \|b_k\|_2 \leqslant b$. Applying the triangle inequality, we have $\|\sum_{k=1}^K U_k b_k\|_2 \leqslant Kb$. Because $U_k b_k \in \textsc{Range}(\bar{U}_K)$ and so is their sum, i.e. $\sum_{k=1}^K U_k b_k \in \textsc{Range}(\bar{U}_K)$, applying (10) yields (8).

Step 5. Let $A_K = \{\sigma_{\min}(\bar{U}_K) \geqslant \sqrt{K} a_K\}$, so $\Pr[A_K] \to 1$ by assumption. On A_K, (10) bounds $\|\Delta_\omega^{K,\text{noise}}\|_2$ pointwise by $\|\sum_k \eta_k\|_2 / (\sqrt{K} a_K)$, so for every $\varepsilon > 0$,

$$\Pr\left[\|\Delta_\omega^{K,\text{noise}}\|_2 > \varepsilon\right] \leqslant \Pr[A_K^c] + \Pr\left[\|\textstyle\sum_k \eta_k\|_2 > \varepsilon \sqrt{K} a_K\right] \leqslant \Pr[A_K^c] + \frac{Cd}{\varepsilon^2 a_K^2} \longrightarrow 0,$$

by Markov's inequality and (7), since $\Pr[A_K] \to 1$ and $a_K \to \infty$. Likewise, on $\{\sigma_{\min}(\bar{U}_K) \geqslant \alpha K\}$ the bound (8) gives $\|\Delta_\omega^{K,\text{bias}}\|_2 \leqslant b/\alpha$ pointwise, so $\Pr[\|\Delta_\omega^{K,\text{bias}}\|_2 > b/\alpha] \leqslant \Pr[\sigma_{\min}(\bar{U}_K) < \alpha K] \to 0$. Finally, if $b = 0$ then $\Delta_\omega^K = \Delta_\omega^{K,\text{noise}}$, and the last claim follows from the first limit. □

Remark 1. *Independence enters Corollary 1 only through the cross terms of Step 3. What the argument needs is that the centered terms are pairwise uncorrelated, and the martingale difference property (9) supplies this without independence. The two proofs differ in one further respect. Corollary 1 treats $\{\theta_k\}$ as a given sequence, so $\bar{U}_K$ is deterministic and the covariance bound $\mathrm{Cov}[\Delta_\omega^K] \preccurlyeq CK(\bar{U}_K^-)^2$ can be stated directly. Here $\bar{U}_K$ is random and statistically coupled with M_K, which is why Step 5 conditions on a spectral event instead.*

Remark 2. *Condition (i) is what restricts the scope of Corollary 2. In Corollary 1, $\{\theta_k\}$ is a given sequence and U_k is deterministic, so the question of which follower model Σ_k is evaluated at does not arise, and our implementation uses the most direct approximation of $\Sigma^{\theta_k\dagger}$, namely the surrogate model built from $\widehat{\omega}_k$. Once the sequence is adaptively generated, (i) requires Σ_k to be computed from θ_k together with the reward estimate available at the end of iteration $k-1$, for instance the surrogate model built from $\bar{\omega}_{k-1}$, or from a trajectory subset disjoint from the one used for $\widehat{\omega}_k$. Corollary 2 therefore covers these variants rather than Algorithm 1 as implemented. Otherwise a term $\mathrm{Cov}(U_k, \widehat{\omega}_k - \omega_k \mid \mathcal{F}_{k-1})$ survives even when $b_k = 0$. This is a requirement introduced by the stronger analysis rather than a defect of the original formulation, and since U_k depends only on the eigenspace partition of Σ_k at the threshold ϵ, the substitution is expected to leave U_k unchanged whenever the spectral gap at ϵ exceeds the estimation perturbation.*

Remark 3. *The spectral hypotheses also differ from that of Corollary 1, which asks for $\sigma_{\max}(\bar{U}_K^-) = o(K^{-1/2})$. The condition $\sigma_{\min}(\bar{U}_K) \geqslant \alpha K$ needed for the bias term is strictly stronger, and coincides with the persistent excitation condition familiar from adaptive control and system identification, where it is either assumed of the input signal or enforced by injecting a dither signal. Empirically, $\mathrm{cond}(\bar{U}_K)$ remains bounded in $[7.3, 103.1]$ across three seeds over 400 outer iterations, which together with $\sigma_{\max}(\bar{U}_K) = \Theta(K)$ implies linear growth of $\sigma_{\min}(\bar{U}_K)$.*

A.5 Proposition 2

Proof. First, we show that $\|r_{\widehat{\omega}} - r_{\omega^\star} - c\|_\infty \leqslant L\|\Delta_\omega\|_2$. It is straightforward from the linear reward structure: for any $(s, b) \in \mathcal{S} \times \mathcal{B}$,

$$\begin{aligned}
|r_{\widehat{\omega}}(s,b) - r_{\omega^\star}(s,b) - c| &= |\langle \widehat{\omega}, \phi(s,b)\rangle - \langle \omega^\star, \phi(s,b)\rangle - c\langle \Delta_{\mathrm{const}}, \phi(s,b)\rangle| \\
&= |\langle \widehat{\omega} - (\omega^\star + c\Delta_{\mathrm{const}}), \phi(s,b)\rangle| \\
&= |\langle \Delta_\omega, \phi(s,b)\rangle| \\
&\leqslant \|\Delta_\omega\|_2 \|\phi(s,b)\|_2 \\
&\leqslant L\|\Delta_\omega\|_2.
\end{aligned}$$

Next, we show that $\|\widehat{V} - V_c^\star\|_\infty \leqslant (1-\gamma_F)^{-1}\|r_{\widehat{\omega}} - (r_{\omega^\star} + c)\|_\infty$, where $V_c^\star$ is the optimal value function corresponding to $r_{\omega^\star} + c$. Let $\mathcal{T}_r$ be the soft Bellman operator corresponding to reward function r, defined as

$$\mathcal{T}_r V = \log\left(\int \exp\left(r(s,b) + \gamma_F \mathbb{E}_{s'}[V(s')]\right) \mathrm{d}b\right),$$

where $\mathbb{E}_{s'}$ is the expectation taken for the next state s' given the current state s and the action b. It has been shown that the soft Bellman operator is γ_F contraction (Haarnoja et al., 2017).

Because $r_{\widehat{\omega}}(s,b) \leqslant r_{\omega^\star}(s,b) + c + \|r_{\widehat{\omega}} - (r_{\omega^\star} + c)\|_\infty$ for all $(s,b) \in \mathcal{S} \times \mathcal{B}$, we have

$$\begin{aligned}
\mathcal{T}_{r_{\widehat{\omega}}} V &= \log\left(\int \exp\left(r_{\widehat{\omega}}(s,b) + \gamma_F \mathbb{E}_{s'}[V(s')]\right) \mathrm{d}b\right) \\
&\leqslant \log\left(\int \exp\left(r^\star(s,b) + c + \|r_{\widehat{\omega}} - (r_{\omega^\star} + c)\|_\infty + \gamma_F \mathbb{E}_{s'}[V(s')]\right) \mathrm{d}b\right) \\
&= \|r_{\widehat{\omega}} - (r_{\omega^\star} + c)\|_\infty + \log\left(\int \exp\left(r^\star(s,b) + c + \gamma_F \mathbb{E}_{s'}[V(s')]\right) \mathrm{d}b\right) \\
&= \|r_{\widehat{\omega}} - (r_{\omega^\star} + c)\|_\infty + \mathcal{T}_{r^\star + c} V.
\end{aligned}$$

Analogously, by using the other side of the inequality, $r_{\widehat{\omega}}(s,b) \geqslant r_{\omega^\star}(s,b) + c - \|r_{\widehat{\omega}} - (r_{\omega^\star} + c)\|_\infty$, we have $\mathcal{T}_{r_{\widehat{\omega}}} V \geqslant -\|r_{\widehat{\omega}} - (r_{\omega^\star} + c)\|_\infty + \mathcal{T}_{r^\star + c} V$. Therefore, for any V, we have

$$\|\mathcal{T}_{r_{\widehat{\omega}}} V - \mathcal{T}_{r^\star + c} V\|_\infty \leqslant \|r_{\widehat{\omega}} - (r_{\omega^\star} + c)\|_\infty.$$

Because soft Bellman operators are contraction maps, $V_c^\star$ and $\widehat{V}$ are the unique fixed points of $\mathcal{T}_{r_{\omega^\star}+c}$ and $\mathcal{T}_{r_{\widehat{\omega}}}$, respectively. Therefore, we have

$$\begin{aligned}\|\widehat{V} - V_c^\star\|_\infty &= \|\mathcal{T}_{r_{\widehat{\omega}}}\widehat{V} - \mathcal{T}_{r_{\omega^\star}+c}V_c^\star\|_\infty \\ &= \|(\mathcal{T}_{r_{\widehat{\omega}}}\widehat{V} - \mathcal{T}_{r_{\omega^\star}+c}\widehat{V}) + (\mathcal{T}_{r_{\omega^\star}+c}\widehat{V} - \mathcal{T}_{r_{\omega^\star}+c}V_c^\star)\|_\infty \\ &\leqslant \|\mathcal{T}_{r_{\widehat{\omega}}}\widehat{V} - \mathcal{T}_{r_{\omega^\star}+c}\widehat{V}\|_\infty + \|\mathcal{T}_{r_{\omega^\star}+c}\widehat{V} - \mathcal{T}_{r_{\omega^\star}+c}V_c^\star\|_\infty \\ &\leqslant \|r_{\widehat{\omega}} - (r_{\omega^\star}+c)\|_\infty + \gamma_F\|\widehat{V} - V_c^\star\|_\infty.\end{aligned}$$

Rearranging the right-most side, we have

$$\|\widehat{V} - V_c^\star\|_\infty \leqslant (1-\gamma_F)^{-1}\|r_{\widehat{\omega}} - (r_{\omega^\star}+c)\|_\infty.$$

The right-hand side of the above inequality is derived by using $\|r_{\widehat{\omega}} - r_{\omega^\star} - c\|_\infty \leqslant L\|\Delta_\omega\|_2$. The left-hand side is derived by using the fact that if the optimal soft value function corresponding to r is V_r, then the optimal soft value function corresponding to $r+c$ is $V_{r+c} = V_r + c(1-\gamma_F)^{-1}$. Hence, we have $\|\widehat{V} - V^\star - c(1-\gamma_F)^{-1}\|_\infty \leqslant L(1-\gamma_F)^{-1}\|\Delta_\omega\|_2$.

Finally, we bound the HG. First, we derive the bound for the Conf-MDP setting. Let $\nabla_\theta J_L(\theta, g^{\theta\dagger})$ be the HG derived in (11), where the ground-truth value function of the follower is used. Let $\nabla_\theta \widehat{J_L(\theta, g^{\theta\dagger})}$ be the HG computed using the estimated value function of the follower, $\widehat{V}$. Then, it is easy to see that only the term $\nabla_\theta Q_F^{\theta\dagger}$ depends on the follower's value function. Therefore,

$$\left\|\nabla_\theta \widehat{J_L(\theta, g^{\theta\dagger})} - \nabla_\theta J_L(\theta, g^{\theta\dagger})\right\| \leqslant \mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma_L^t\frac{|B_L^{\theta\dagger}(s_t,b_t)|}{\beta}\left\|\nabla_\theta Q_F^{\theta\dagger}(s_t,b_t) - \nabla_\theta\widehat{Q_F^{\theta\dagger}(s_t,b_t)}\right\|\right],$$

where $\nabla_\theta Q_F^{\theta\dagger}(s_t,b_t)$ and $\nabla_\theta\widehat{Q_F^{\theta\dagger}(s_t,b_t)}$ are as defined in (6) with the ground-truth value function $V^\star$ and its estimate $\widehat{V}$. Noting that for any constant $C \in \mathbb{R}$,

$$\begin{aligned}&\mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma_F^{t+1}C\nabla_\theta\log p^\theta(s_{t+1}\mid s_t,b_t)\,\middle|\,\begin{matrix}s_0=s\\b_0=b\end{matrix}\right] \\ &\quad= \sum_{t=0}^{\infty}\gamma_F^{t+1}\mathbb{E}_\tau\left[C\nabla_\theta\log p^\theta(s_{t+1}\mid s_t,b_t)\,\middle|\,\begin{matrix}s_0=s\\b_0=b\end{matrix}\right] \\ &\quad= \sum_{t=0}^{\infty}\gamma_F^{t+1}\mathbb{E}_\tau\left[\mathbb{E}_t[C\nabla_\theta\log p^\theta(s_{t+1}\mid s_t,b_t)]\,\middle|\,\begin{matrix}s_0=s\\b_0=b\end{matrix}\right] \\ &\quad= \sum_{t=0}^{\infty}\gamma_F^{t+1}\mathbb{E}_\tau\left[\nabla_\theta\mathbb{E}_t[C]\,\middle|\,\begin{matrix}s_0=s\\b_0=b\end{matrix}\right] \\ &\quad= 0,\end{aligned}$$

where $\mathbb{E}_t$ denotes the conditional expectation under the natural filtration generated by $\{(s_i,b_i)\}_{i\leqslant t}$, we have

$$\nabla_\theta Q_F^{\theta\dagger}(s,b) = \mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma_F^{t+1}(V_F^{\theta\dagger}(s_{t+1})+C)\nabla_\theta\log p^\theta(s_{t+1}\mid s_t,b_t)\,\middle|\,\begin{matrix}s_0=s\\b_0=b\end{matrix}\right].$$

With $C = c(1-\gamma_F)^{-1}$, we have

$$
\begin{aligned}
&\left\|\nabla_\theta Q_F^{\theta\dagger}(s,b) - \nabla_\theta \widehat{Q_F^{\theta\dagger}(s,b)}\right\| \\
&\quad= \left\|\mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma_F^{t+1}(V^\star(s_{t+1}) + c(1-\gamma_F)^{-1} - \widehat{V}(s_{t+1}))\nabla_\theta \log p^\theta(s_{t+1}\mid s_t,b_t)\,\middle|\, \begin{array}{l} s_0=s \\ b_0=b \end{array}\right]\right\| \\
&\quad\leqslant \mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma_F^{t+1}\left|V^\star(s_{t+1}) + c(1-\gamma_F)^{-1} - \widehat{V}(s_{t+1})\right|\left\|\nabla_\theta \log p^\theta(s_{t+1}\mid s_t,b_t)\right\|\,\middle|\, \begin{array}{l} s_0=s \\ b_0=b \end{array}\right] \\
&\quad\leqslant \|\widehat{V} - V_c^\star\|_\infty \mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma_F^{t+1}\left\|\nabla_\theta \log p^\theta(s_{t+1}\mid s_t,b_t)\right\|\,\middle|\, \begin{array}{l} s_0=s \\ b_0=b \end{array}\right] \\
&\quad\leqslant L(1-\gamma_F)^{-1}\|\Delta_\omega\|_2 \mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma_F^{t+1}\left\|\nabla_\theta \log p^\theta(s_{t+1}\mid s_t,b_t)\right\|\,\middle|\, \begin{array}{l} s_0=s \\ b_0=b \end{array}\right].
\end{aligned}
$$

Hence, we have

$$
\begin{aligned}
&\left\|\nabla_\theta \widehat{J_L(\theta, g^{\theta\dagger})} - \nabla_\theta J_L(\theta, g^{\theta\dagger})\right\| \\
&\quad\leqslant \frac{L\|\Delta_\omega\|_2}{1-\gamma_F}\mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma_L^t \frac{|B_L^{\theta\dagger}(s_t,b_t)|}{\beta}\mathbb{E}_{\tilde{\tau}}\left[\sum_{\tilde{t}=0}^{\infty}\gamma_F^{\tilde{t}+1}\left\|\nabla_\theta \log p^\theta(\tilde{s}_{\tilde{t}+1}\mid \tilde{s}_{\tilde{t}},\tilde{b}_{\tilde{t}})\right\|\,\middle|\, \begin{array}{l} \tilde{s}_0=s_t \\ \tilde{b}_0=b_t \end{array}\right]\right].
\end{aligned}
$$

This completes the proof for the Conf-MDP case.

Next, we derive the bound for 2-player MGs. Similarly to the above argument, we have from (12)

$$
\begin{aligned}
&\left\|\nabla_\theta \widehat{J_L(\theta, g^{\theta\dagger})} - \nabla_\theta J_L(\theta, g^{\theta\dagger})\right\| \\
&\qquad\qquad\leqslant \frac{1}{1-\gamma_L}\left\|\mathbb{E}\left[\frac{B_L^{\theta\dagger}(s,a,b)}{\beta}\left(\nabla_\theta Q_F^{\theta\dagger}(s,a,b) - \nabla_\theta \widehat{Q_F^{\theta\dagger}(s,a,b)}\right)\right]\right\|,
\end{aligned}
$$

where $\nabla_\theta Q_F^{\theta\dagger}(s,a,b)$ and $\nabla_\theta \widehat{Q_F^{\theta\dagger}(s,a,b)}$ are as defined in (13) with the ground-truth value function $V^\star$ and its estimate $\widehat{V}$, and the expectation is taken for $(s,a,b) \sim g^{\theta\dagger}(b \mid s,a) f_\theta(a \mid s) d_{\gamma_L}^{\theta\dagger}(s)$. Noting that for any constant $C \in \mathbb{R}$,

$$
\begin{aligned}
&\mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma_F^t C\nabla_\theta \log f_\theta(a_t\mid s_t)\,\middle|\, \begin{array}{l} s_0=s \\ a_0=a \\ b_0=b \end{array}\right] \\
&\quad= \sum_{t=0}^{\infty}\gamma_F^{t+1}\mathbb{E}_\tau\left[C\nabla_\theta \log f_\theta(a_t\mid s_t)\,\middle|\, \begin{array}{l} s_0=s \\ a_0=a \\ b_0=b \end{array}\right] \\
&\quad= C\nabla_\theta \log f_\theta(a\mid s) + \sum_{t=1}^{\infty}\gamma_F^{t+1}\mathbb{E}_\tau\left[\mathbb{E}_t[C\nabla_\theta\nabla_\theta \log f_\theta(a_t\mid s_t)\,\middle|\, \begin{array}{l} s_0=s \\ a_0=a \\ b_0=b \end{array}\right] \\
&\quad= C\nabla_\theta \log f_\theta(a\mid s) + \sum_{t=1}^{\infty}\gamma_F^{t+1}\mathbb{E}_\tau\left[\nabla_\theta\mathbb{E}_t[C]\,\middle|\, \begin{array}{l} s_0=s \\ a_0=a \\ b_0=b \end{array}\right] \\
&\quad= C\nabla_\theta \log f_\theta(a\mid s),
\end{aligned}
$$

where $\mathbb{E}_t$ denotes the conditional expectation under the natural filtration generated by $\{(s_i,a_i,b_i)\}_{i\leqslant t}$, we have

$$
\nabla_\theta Q_F^{\theta\dagger}(s,a,b) = \mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma_F^{t+1}(V_F^{\theta\dagger}(s_{t+1},a_{t+1}) + C)\nabla_\theta \log f_\theta(a_t\mid s_t)\,\middle|\, \begin{array}{l} s_0=s \\ a_0=a \\ b_0=b \end{array}\right] - C\nabla_\theta \log f_\theta(a\mid s).
$$

Noting that

$$\mathbb{E}\left[\frac{B_L^{\theta\dagger}(s,a,b)}{\beta}C\nabla_\theta \log f_\theta(a \mid s)\right] = \mathbb{E}\left[\frac{\mathbb{E}_{b\sim g^{\theta\dagger}}[B_L^{\theta\dagger}(s,a,b)]}{\beta}C\nabla_\theta \log f_\theta(a \mid s)\right] = 0,$$

we have with $C = c(1-\gamma_F)^{-1}$,

$$\begin{aligned}
&\left\|\mathbb{E}\left[\frac{B_L^{\theta\dagger}(s,a,b)}{\beta}\left(\nabla_\theta Q_F^{\theta\dagger}(s,a,b) - \nabla_\theta \widehat{Q_F^{\theta\dagger}(s,a,b)}\right)\right]\right\| \\
&= \left\|\mathbb{E}\left[\frac{B_L^{\theta\dagger}(s,a,b)}{\beta}\mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma_F^{t+1}(V^\star(s_{t+1}) + \frac{c}{1-\gamma_F} - \widehat{V}(s_{t+1}))\nabla_\theta \log f_\theta(a_t \mid s_t)\,\middle|\, \begin{array}{l} s_0=s \\ a_0=a \\ b_0=b \end{array}\right]\right]\right\| \\
&\leqslant \mathbb{E}\left[\frac{|B_L^{\theta\dagger}(s,a,b)|}{\beta}\mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma_F^{t+1}|V^\star(s_{t+1}) + \frac{c}{1-\gamma_F} - \widehat{V}(s_{t+1})|\|\nabla_\theta \log f_\theta(a_t \mid s_t)\|\,\middle|\, \begin{array}{l} s_0=s \\ a_0=a \\ b_0=b \end{array}\right]\right] \\
&\leqslant \|\widehat{V} - V_c^\star\|_\infty \mathbb{E}\left[\frac{|B_L^{\theta\dagger}(s,a,b)|}{\beta}\mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma_F^{t+1}\|\nabla_\theta \log f_\theta(a_t \mid s_t)\|\,\middle|\, \begin{array}{l} s_0=s \\ a_0=a \\ b_0=b \end{array}\right]\right].
\end{aligned}$$

The rest follows the same argument as the Conf-MDP case. This completes the proof. □

B Detail Implementation

B.1 Hypergradient derived in Kudo et al. (2026)

The HG in Conf-MDP settings can be explicitly written as

$$\nabla_\theta J_L(\theta, g^{\theta\dagger}) = \mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma_L^t\left(\nabla_\theta r_L^\theta(s_t,b_t) + V_L^{\theta\dagger}(s_t)\nabla_\theta \log p^\theta(s_t \mid s_{t-1}, b_{t-1}) + \frac{B_L^{\theta\dagger}(s_t,b_t)}{\beta}\nabla_\theta Q_F^{\theta\dagger}(s_t,b_t)\right)\right], \quad (11)$$

where $p^\theta(s_0 \mid s_{-1}, b_{-1})$ is interpreted as the initial distribution $\rho_0^\theta(s_0)$.

In the context of 2-player MGs, the benefit of the leader f_θ, given that the follower takes action b at state s under the leader's action a, is defined as:

$$B_L^{\theta\dagger}(s,a,b) := Q_L^{\theta\dagger}(s,a,b) - \mathbb{E}_{b\sim g^{\theta\dagger}(\cdot|s,a)}\left[Q_L^{\theta\dagger}(s,a,b)\right].$$

The HG for 2-player MG with a stochastic leader policy is derived as follows. In the following, $d_{\gamma_L}^{\theta\dagger}(s)$ denotes the discounted state visitation distribution under the leader's policy f_θ and its corresponding best response follower's policy $g^{\theta\dagger}$.

Kudo et al. (2026) show that the HG of (4) with respect to θ is given by

$$\nabla_\theta J_L(\theta, g^{\theta\dagger}) = \frac{1}{1-\gamma_L}\mathbb{E}\left[Q_L^{\theta\dagger}(s,a,b)\nabla_\theta \log f_\theta(a \mid s) + \frac{B_L^{\theta\dagger}(s,a,b)}{\beta}\nabla_\theta Q_F^{\theta\dagger}(s,a,b)\right], \quad (12)$$

where the expectation $\mathbb{E}$ is taken for $(s,a,b) \sim g^{\theta\dagger}(b \mid s,a)f_\theta(a \mid s)d_{\gamma_L}^{\theta\dagger}(s)$, and

$$\nabla_\theta Q_F^{\theta\dagger}(s,a,b) = \mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma_F^t V_F^{\theta\dagger}(s_t,a_t)\nabla_\theta \log f_\theta(a_t \mid s_t)\,\middle|\, \begin{array}{l} s_0=s \\ a_0=a \\ b_0=b \end{array}\right]. \quad (13)$$

B.2 Hypergradient Baseline

B.2.1 Hypergradient Baselines for Conf-MDP

Based on the HG formulation derived in Appendix B.1, we introduce baseline functions to reduce the variance of the gradient estimators. For the Conf-MDP setting in Equation (11), we subtract

$\nabla_\theta V_F^{\theta\dagger}(s_t, b_t)$ from $\nabla_\theta Q_F^{\theta\dagger}(s_t, b_t)$. Because $\nabla_\theta V_F$ is independent of the action b, it effectively serves as an unbiased baseline when marginalized over the action distribution:

$$\nabla_\theta J_L(\theta, g^{\theta\dagger}) = \mathbb{E}_\tau \left[\sum_{t=0}^{\infty} \gamma_L^t \Big(\nabla_\theta r_L^\theta(s_t, b_t) + V_L^{\theta\dagger}(s_t) \nabla_\theta \log p^\theta(s_t \mid s_{t-1}, b_{t-1}) + \frac{B_L^{\theta\dagger}(s_t, b_t)}{\beta} (\nabla_\theta Q_F^{\theta\dagger}(s_t, b_t) - \nabla_\theta V_F^{\theta\dagger}(s_t)) \Big) \right],$$

where we have $\nabla_\theta V_F^{\theta\dagger}(s_t) = \mathbb{E}_{g^{\theta\dagger}}[\nabla_\theta Q_F^{\theta\dagger}(s_t, b_t) \mid s_t]$. Note that this formulation is not used in Kudo et al. (2026) because estimating both $\nabla_\theta Q_F^{\theta\dagger}(s_t, b_t)$ and its expectation $\nabla_\theta V_F^{\theta\dagger}(s_t)$ requires multiple trajectories starting from the same state s_t. In our setting, we can easily approximate $\nabla_\theta V_F^{\theta\dagger}(s_t)$ by using the surrogate follower $\widehat{g}$.

To further reduce variance, an alternative formulation incorporating a temporal difference-like baseline can be employed by subtracting $V_L^{\theta\dagger}(s_{t+1})$ from $V_L^{\theta\dagger}(s_t)$:

$$\nabla_\theta J_L(\theta, g^{\theta\dagger}) \approx \mathbb{E}_\tau \left[\sum_{t=0}^{\infty} \gamma_L^t \Big(\nabla_\theta r_L^\theta(s_t, b_t) + (V_L^{\theta\dagger}(s_t) - \mathbb{E}[V_L^{\theta\dagger}(s_{t+1})]) \nabla_\theta \log p^\theta(s_t \mid s_{t-1}, b_{t-1}) + \frac{B_L^{\theta\dagger}(s_t, b_t)}{\beta} (\nabla_\theta Q_F^{\theta\dagger}(s_t, b_t) - \nabla_\theta V_F^{\theta\dagger}(s_t)) \Big) \right]. \quad (14)$$

However, it is important to note that this specific TD-like term does not act as a strict baseline, making the resulting gradient estimator biased rather than unbiased.

B.2.2 Hypergradient Baselines for 2-player MG

Similarly, for the 2-player MG setting derived in (12), we apply a baseline to reduce variance by subtracting $\nabla_\theta V_F^{\theta\dagger}(s, a)$ from $\nabla_\theta Q_F^{\theta\dagger}(s, a, b)$:

$$\nabla_\theta J_L(\theta, g^{\theta\dagger}) = \frac{1}{1-\gamma_L} \mathbb{E} \left[Q_L^{\theta\dagger}(s, a, b) \nabla_\theta \log f_\theta(a \mid s) + \frac{B_L^{\theta\dagger}(s, a, b)}{\beta} (\nabla_\theta Q_F^{\theta\dagger}(s, a, b) - \nabla_\theta V_F^{\theta\dagger}(s, a)) \right]. \quad (15)$$

The baseline formulation utilized in this setting is strictly unbiased, meaning it yields an unbiased gradient estimator.

B.3 Reward Identification Algorithm

Algorithm 2 Reward Identification

Input: ω_k, Σ_k,v_{k-1}, $\bar{U}_{k-1}$, threshold parameter $\epsilon \geqslant 0$.
1: Perform eigen decomposition $E\Lambda E^\top = \Sigma_k$
2: Compute a diagonal matrix $I_{\Lambda \succ \epsilon}$, where $[I_{\Lambda \succ \epsilon}]_{dd} = 1$ if $[\Lambda]_{dd} > \epsilon$, 0 otherwise
3: Compute the scaled covariance matrix $U_k = E I_{\Lambda \succ \epsilon} E^\top$
4: $\bar{U}_k \leftarrow \bar{U}_{k-1} + U_k$, $v_k \leftarrow v_{k-1} + U_k \omega_k$
5: Solve $\bar{U}_k \omega = v_k$ to obtain $\bar{\omega}_k$
Output: $\bar{\omega}_k$, v_k, $\bar{U}_k$

B.4 Modules within Proposed Algorithm 1

Note that $\bar{s}$ denotes state s in Conf-MDP settings and denotes a pair of state s and leader action a in 2-player MGs.

Algorithm 3 `CriticUpdate`$(Q_L^i, \mathcal{R})$ (Expected SARSA variant)

Input: Current critic Q_L^i, policies $f, \widehat{g}$, batch $\mathcal{R}$, learning rate α, discount factor γ_L

1: **for** $(\bar{s}, b, r_L, \bar{s}') \in \mathcal{R}$ **do**

2: Compute expected value: $V_L(\bar{s}') \leftarrow \sum_{a'} f(a'|s') \sum_{b'} \widehat{g}(b'|\bar{s}') Q_L^i(\bar{s}', b')$

3: $y \leftarrow r_L + \gamma_L V_L(\bar{s}')$

4: $Q_L^i(\bar{s}, b) \leftarrow Q_L^i(\bar{s}, b) + \alpha \left(y - Q_L^i(\bar{s}, b)\right)$

5: $Q_L^{i+1} \leftarrow Q_L^i$

Output: Q_L^{i+1}

Algorithm 4 `CriticUpdate` (TD with target network)

Input: Online critic Q_{ω_L}, target critic $Q_{\bar{\omega}_L}$, batch $\mathcal{R}$, discount γ_L, learning rate α, Polyak factor τ

1: **for** $(\bar{s}, b, r_L, \bar{s}', b', d) \in \mathcal{R}$ **do**

2: $y \leftarrow r_L + \gamma_L (1 - d)\, Q_{\bar{\omega}_L}(\bar{s}', b')$

3: $\omega_L \leftarrow \omega_L - \alpha \nabla_{\omega_L} \frac{1}{N} \sum_{j=1}^{N} \Big(Q_{\omega_L}(\bar{s}_j, b_j) - y_j \Big)^2$

4: $\bar{\omega}_L \leftarrow (1 - \tau)\bar{\omega}_L + \tau\, \omega_L$ (Polyak averaging)

Output: Updated critics $Q_{\omega_L}, Q_{\bar{\omega}_L}$.

Algorithm 5 `PartialDerivative`$(\mathcal{R}, \theta^i, V_L, \cdot)$ in Conf-MDP Settings

Input: Trajectory batch $\mathcal{R}$, current leader parameter θ^i, leader value function V_L, initial state distribution ρ_0

1: $\widehat{\partial_\theta J_L} \leftarrow \frac{1}{|\mathcal{R}|} \sum_{\tau \in \mathcal{R}} \Bigg(V_L(s_0) \nabla_\theta \log \rho_0(s_0) + \sum_{t=0}^{T-1} \Big(\gamma_L^t \nabla_\theta r_L^{\theta^i}(s_t, b_t) + \gamma_L^{t+1} V_L(s_{t+1}) \nabla_\theta \log p^{\theta^i}(s_{t+1} \mid s_t, b_t) \Big) \Bigg)$

Output: $\widehat{\partial_\theta J_L}$

Algorithm 6 `PartialDerivative`$(\mathcal{R}, \theta^i, \cdot, Q_L)$ in Markov Game Settings

Input: Transition sample batch $\mathcal{R}$, current leader parameter θ^i, leader Q-function Q_L, leader policy f_{θ^i}, surrogate follower policy $\widehat{g}$

1: $V_L(s) \leftarrow \sum_{a'} f_{\theta^i}(a'|s) \sum_{b'} \widehat{g}(b'|s, a')\, Q_L(s, a', b')$

2: $\widehat{\partial_\theta J_L} \leftarrow \frac{1}{|\mathcal{R}|} \sum_{(s,a,b) \in \mathcal{R}} \big(Q_L(s, a, b) - V_L(s) \big) \nabla_\theta \log f_{\theta^i}(a \mid s)$ # Empirical approximation of eq. (12) without discount factor

Output: $\widehat{\partial_\theta J_L}$

Algorithm 7 `FollowerQGrad`$(s, a, b, \mathcal{R}, \theta^i, V_F)$ in Markov Game Settings

Input: Target state s, leader action a, follower action b, full trajectory batch $\mathcal{R}$, current leader parameter θ^i, follower soft value V_F

1: Extract segments: $\mathcal{R}_{sab} \leftarrow \left\{ \tau^{k:T} \mid (s_k, a_k, b_k) = (s, a, b) \text{ for } (s_k, a_k, b_k) \in \tau \in \mathcal{R} \right\}$

2: $\widehat{\nabla_\theta Q_F}(s, a, b) \leftarrow \frac{1}{|\mathcal{R}_{sab}|} \sum_{\tau^{k:T} \in \mathcal{R}_{sab}} \sum_{t=k+1}^{T} \gamma_F^{t-k} V_F(s_t, a_t)$

Output: $\widehat{\nabla_\theta Q_F}(s, a, b)$

Algorithm 8 Influence Term

1: $\widehat{\nabla_{\theta_k} J_L^k} \leftarrow \widehat{\partial_{\theta_k} J_L^k} + \frac{1}{|\mathcal{R}_k|} \sum_{(s,a,b)\in\mathcal{R}_k} B_L^{k+1}(s,a,b) \widehat{\nabla_{\theta_k} Q_F^k}(s,a,b)$

Algorithm 9 Influence Term with Baseline in Markov Game Settings (Discrete Case)

1: Compute baseline: $\widehat{\nabla_{\theta_k} V_F^k}(s,a) \leftarrow \sum_{b'} \hat{g}_k(b' \mid s,a) \widehat{\nabla_{\theta_k} Q_F^k}(s,a,b')$
2: $\widehat{\nabla_{\theta_k} J_L^k} \leftarrow \widehat{\partial_{\theta_k} J_L^k} + \frac{1}{|\mathcal{R}_k|} \sum_{(s,a,b)\in\mathcal{R}_k} B_L^{k+1}(s,a,b) \left(\widehat{\nabla_{\theta_k} Q_F^k}(s,a,b) - \widehat{\nabla_{\theta_k} V_F^k}(s,a) \right)$ # Based on eq. (15)

C Experiment

C.1 Implementation Details

Table 2 summarizes the implementation choices used in each environment. While the overall algorithmic framework is shared across settings, several components are instantiated differently depending on whether the transition dynamics are known or unknown. In the Conf-MDP experiments, quantities such as $Q_F^{\theta\dagger}$, $Q_L^{\theta\dagger}$, $V_L^{\theta\dagger}$, and $\nabla Q_F^{\theta\dagger}$ are computed analytically from the known system dynamics using the Discrete Algebraic Riccati Equation (DARE) and discrete Lyapunov equations. In the 2-player Markov game experiments, surrogate policies and value-related terms are computed either by dynamic programming or by successor-feature-based estimation, depending on whether the transition model is available. For the continuous setting, the critic is first pre-trained with SARSA-TD updates from randomly initialized trajectories, and the transition dynamics are estimated by linear regression when unknown.

Table 2: Implementation Details by Environment

Setting	Component	Algorithm 1 Line	Implementation Details
Conf-MDP (Common)	$Q_F^{\theta\dagger}$	7	Since the transition dynamics are linear-Gaussian and the reward functions are quadratic, it is derived directly from the DARE solution.
	$Q_L^{\theta\dagger}, V_L^{\theta\dagger}$	9	Analytically computed by solving the DARE.
	$\nabla Q_F^{\theta\dagger}$	13	Obtained via the discrete Lyapunov equation.
Conf-MDP (Easy)	HG Calculation	15	Computed analytically using the known transition dynamics and policies, which bypasses the need for empirical trajectory sampling.
Conf-MDP (Hard)	HG Calculation	15	A variance reduction baseline is adopted to stabilize stochastic gradient estimates, using the expectation $\mathbb{E}[V_L^{\theta\dagger}(s_{t+1})]$ obtained via DARE as a baseline for the $\nabla_\theta \log p^\theta$ term.
2-player MG (Discrete)	Surrogate Policy $\hat{g}_k$	5, 6	Computed via Soft Value Iteration (SoftVI) if the transition probability (P) is known, or by incorporating Successor Features (SFs) to calculate expected feature values if unknown.
	Value functions & gradients	9, 10, 12, 13	If P is known, computed using dynamic programming to eliminate sampling noise and improve learning efficiency.
2-player MG (Continuous)	Critic (Q_L) Initialization	1	Prior to the main loop, pre-trained for 50 optimization epochs via SARSA-TD updates using 40 episodes (approx. 4,000 transitions) generated from a randomized initial distribution.
	Transition Dynamics	5	Estimated via linear regression if unknown.
	Benefit (B_L)	12	Calculated as the difference between the online leader's action-value function (Q_L^{online}) and the target leader's value function (V_L^{target}).
	$\nabla Q_F^{\theta\dagger}$	13	Obtained via the discrete Lyapunov equation.

C.2 Synthetic task to evaluate the effect of shaped reward.

We verify that a single IRL reward bias cannot achieve the true optimal objective in a simple environment due to dynamics mismatch.

We consider a Conf-MDP $\mathcal{M}_\theta$, where the state space is $\mathcal{S} = \{s_1, s_2\}$ and the action space is $\mathcal{B} = \{b_1, b_2\}$. The true follower's reward is $r_F(s, b) = 0$ for all (s, b). The transition probabilities are parameterized by $\theta \in (0, 1)$ such that $p(s_1 \mid s_1, b_1) = p(s_1 \mid s_2, b_2) = \theta$, $p(s_1 \mid s_1, b_2) = 1$, and $p(s_1 \mid s_2, b_1) = 0$.

Instead of a standard environment reward, we define a diagnostic meta-reward to explicitly evaluate the behavior of the surrogate policy. Let the leader's reward be $r_L = -(0.5 - \hat{g}(b_t \mid s_t))^2$ and the regularization term be $\Omega(\theta) = -(\theta - 0.5)^2$, where $\hat{g}$ is the estimated follower policy. The leader's diagnostic meta-objective is defined as:

$$J_L(\theta, g^{\theta\dagger}) := \mathbb{E}_\tau \left[\sum_{t=0}^{\infty} \gamma_L^t r_L(s_t, b_t) \right] + \Omega(\theta).$$

To simulate an estimated reward vector produced by IRL, we initialize the reward vector as $\mathcal{N}([a, a, 0, 0], I)$ with $a = 1.5$. By observing the gap between the estimated single reward and the aggregate rewards, we evaluate how this estimation error influences the optimization goal. The objective of this experiment is to drive the KL divergence to a uniform distribution to zero, and to converge θ to 0.5.

Since the leader knows the true dynamics, the surrogate follower's policy is recovered using Soft Value Iteration. As shown in the Figures 4 to 6 below, the estimation errors prevent the method from achieving the true optimal objective, confirming our observation.

Results The reward identifiability is not only theoretically but also practically important. The experimental results are shown in Figures 4 to 6. RE-HG effectively eliminates the potential shaping term, which is evidenced by the convergence of Terms 2, 3 in (11) and the estimated reward function to 0. Furthermore, the leader's parameter θ successfully converges to the target value of 0.5. In contrast, the single reward baseline (Single) fails to achieve the optimal objective due to the influence of potential shaping. Specifically, the KL divergence, individual evaluation terms, and the parameters θ converge to sub-optimal values. These results empirically demonstrate that the inability to vanish the potential shaping term inevitably leads to estimation errors when a dynamics mismatch exists.

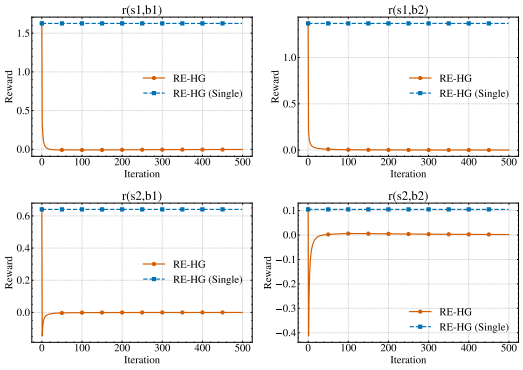


Figure 4: Estimated Rewards

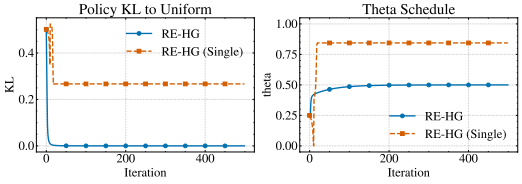


Figure 5: Summary of Diagnostics

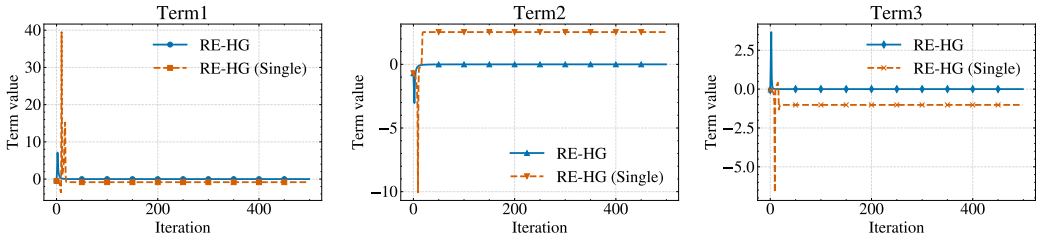


Figure 6: Summary of Terms

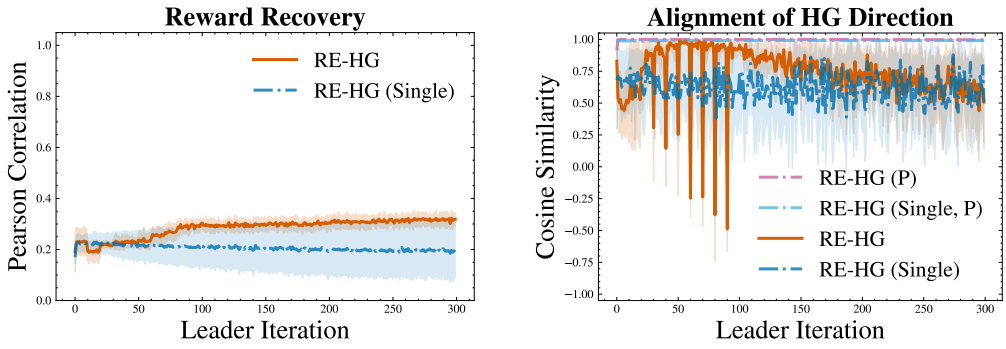
Hyperparameters Hyperparameters used in this experiment are summarized in Table 3.

Table 3: Hyperparameters

Parameter	Symbol	Value
Leader Discount Factor	γ_L	0.95
Follower Discount Factor	γ_F	0.95
Leader Parameter Learning Rate	$\alpha_{L,\text{actor}}$	1.0×10^{-2}
Initial Leader Parameter	θ_0	0.25
Leader Iterations	-	500
Temperature (β)	-	2.5×10^{-1}

C.3 2-player Markov Game (Discrete Case)

Implementation Details In Lines 5, 6 of Algorithm 1, when the transition dynamics P is known, the leader computes the surrogate follower's policy via Soft Value Iteration (SoftVI). At Lines 5, 6 of Algorithm 1, when the transition dynamics P is not known, the leader computes the expected feature value under the initial distribution ρ_0 by incorporating Successor Features (SFs) (Barreto et al., 2017). IRL is performed at iterations 0, 10, ..., 90. At Lines 9, 10, 12, 13 of Algorithm 1, when the transition dynamics P is known, the leader computes the value functions and gradients via dynamic programming. This approach eliminates sampling noise and significantly improves learning efficiency by avoiding Monte-Carlo simulations. However, it should be noted that this is an approximation based on learned SFs rather than unbiased sampling; therefore, the estimates may inherit biases from TD errors during the SF training process. At Line 16 of Algorithm 1, we utilize the Algorithm 9.



(a) Reward Correlation

(b) HG Similarity

Figure 7: Discrete toy task support material."P" indicates the known transition probability scenario. (a) Pearson correlation between the recovered and true rewards, weighted by the undiscounted state-action occupancy measure of the surrogate follower policy. For each run, performance is evaluated as the mean of 10 test rollouts. Avg. (line) and std. (shaded area) across the 3 independent training runs are displayed. (b) Cosine similarity of the HG between the estimated and Oracle models. For each run, performance is evaluated as the mean of 10 test rollouts. Avg. (line) and std. (shaded area) across the 3 independent training runs are displayed.

Result As discussed in Section 5, Figure 7 provides the detailed trajectories of the reward correlation and the HG cosine similarity. Figure 7a shows that the Pearson correlation remains relatively low. This is primarily due to estimation errors in regions where insufficient trajectories were collected, which limits the accuracy of reward recovery. However, as illustrated in Figure 7b, the cosine similarity of the HG remains highly stable throughout the training process. This indicates that the estimation errors in those rarely visited regions do not negatively affect the overall gradient direction.

Hyperparameters Hyperparameters used in this experiment are summarized in Table 4.

Table 4: Hyperparameters

Parameter	Symbol	Value
Leader Discount Factor	γ_L	0.99
Leader Actor Learning Rate	$\alpha_{L,\text{actor}}$	1.0×10^{-2}
Leader Critic Learning Rate	$\alpha_{L,\text{critic}}$	1.0×10^{-3}
Leader Iterations	-	300
Critic Updates per Iteration	-	10
Follower Discount Factor	γ_F	0.9
Follower Learning Rate (SoftQL)	α_F	0.01
Temperature (β)	-	2.5×10^{-1}
Max Inner Iterations	-	150
Inner Episodes	-	50
Successor feature Update Episodes	-	100
Expert Episodes per Iteration	-	200
IRL Rollout Episodes	-	100
ω Learning Rate	-	5×10^{-3}
IRL Timing	-	Step 0,10, ... ,90
Successor feature Learning Rate	α_ω	0.05
Adam First Moment Coefficient	β_1	0.9
Adam Second Moment Coefficient	β_2	0.999
threshold	ϵ	10^{-23}

C.4 Configurable MDP Continuous

Analytical Derivation of the Leader's Value Functions First, we define the system dynamics, the leader's immediate reward, and the follower's best response policy. The state transition is given by:

$$s' = A_\theta s + Bb + w, \quad w \sim \mathcal{N}(\mathbf{0}, \Sigma_w). \tag{16}$$

The leader's immediate reward is defined using the weighting matrices W_L and U_L, and a configuration cost $c_L(\theta)$:

$$r_L(s, b) = -s^\top W_L s - b^\top U_L b - c_L(\theta). \tag{17}$$

The follower's best response policy is $b \sim \mathcal{N}(-K_s^\theta s, \Sigma_F^\theta)$, which can be reparameterized using the noise term ϵ_b as:

$$b = -K^\theta s + \epsilon_b, \quad \epsilon_b \sim \mathcal{N}(\mathbf{0}, \Sigma_F^\theta). \tag{18}$$

We assume the leader's optimal value function takes a quadratic form:

$$V_L^{\theta\dagger}(s) = -s^\top P_L^\theta s - v_L^\theta, \tag{19}$$

where $P_L^\theta \geq 0$ is a symmetric matrix and v_L^θ is a scalar. The leader's action-value function $Q_L^{\theta\dagger}(s, b)$ and the state-value function $V_L^{\theta\dagger}(s)$ satisfy the following Bellman equations:

$$Q_L^{\theta\dagger}(s, b) = r_L(s, b) + \gamma_L t \mathbb{E}_w \left[V_L^{\theta\dagger}(s') \right], \tag{20}$$

$$V_L^{\theta\dagger}(s) = \mathbb{E}_{\epsilon_b} \left[Q_L^{\theta\dagger}(s, b) \right]. \tag{21}$$

To identify P_L^θ and v_L^θ, we first evaluate the expectation over the unobservable disturbance w in Eq. (20). Substituting Eq. (16) and Eq. (19) into Eq. (20), and utilizing the properties $\mathbb{E}_w[w] = \mathbf{0}$ and $\mathbb{E}_w[ww^\top] = \Sigma_w$, we obtain:

$$\begin{aligned}
\mathbb{E}_w \left[V_L^{\theta\dagger}(A_\theta s + Bb + w) \right] &= -\mathbb{E}_w \left[(A_\theta s + Bb + w)^\top P_L^\theta (A_\theta s + Bb + w) \right] - v_L^\theta \\
&= -(A_\theta s + Bb)^\top P_L^\theta (A_\theta s + Bb) - \mathbb{E}_w[\mathrm{Tr}(w^\top P_L^\theta w)] - v_L^\theta \\
&= -(A_\theta s + Bb)^\top P_L^\theta (A_\theta s + Bb) - \mathrm{Tr}(P_L^\theta \Sigma_w) - v_L^\theta.
\end{aligned} \tag{22}$$

Substituting Eq. (17) and Eq. (22) into Eq. (20) yields the closed quadratic form for $Q_L^{\theta\dagger}(s,b)$:

$$Q_L^{\theta\dagger}(s,b) = -s^\top W_L s - b^\top U_L b - c_L(\theta) - \gamma_L((A_\theta s + Bb)^\top P_L^\theta (A_\theta s + Bb) + \mathrm{Tr}(P_L^\theta \Sigma_w) + v_L^\theta). \quad (23)$$

Next, we evaluate Eq. (21) by taking the expectation of Eq. (23) over the follower's action noise ϵ_b. Defining the closed-loop transition matrix as $\tilde{A}_\theta := A_\theta - BK_s^\theta$ and substituting Eq. (18), we expand the state-action terms:

$$V_L^{\theta\dagger}(s) = \mathbb{E}_{\epsilon_b}\Big[- s^\top W_L s - (-K_s^\theta s + \epsilon_b)^\top U_L (-K_s^\theta s + \epsilon_b) - c_L(\theta) \\ - \gamma_L((\tilde{A}_\theta s + B\epsilon_b)^\top P_L^\theta (\tilde{A}_\theta s + B\epsilon_b) + \mathrm{Tr}(P_L^\theta \Sigma_w) + v_L^\theta)\Big].$$

Since $\mathbb{E}_{\epsilon_b}[\epsilon_b] = \mathbf{0}$, the cross terms vanish. By applying $\mathbb{E}_{\epsilon_b}[\epsilon_b \epsilon_b^\top] = \Sigma_F^\theta$, the quadratic terms in ϵ_b evaluate to traces:

$$V_L^{\theta\dagger}(s) = -s^\top \left(W_L + (K_s^\theta)^\top U_L K_s^\theta + \gamma_L \tilde{A}_\theta^\top P_L^\theta \tilde{A}_\theta \right) s \\ - \mathrm{Tr}(U_L \Sigma_F^\theta) - c_L(\theta) - \gamma_L(\mathrm{Tr}(P_L^\theta B \Sigma_F^\theta B^\top) + \mathrm{Tr}(P_L^\theta \Sigma_w) + v_L^\theta). \quad (24)$$

Finally, by matching the quadratic and scalar coefficients of Eq. (24) with the ansatz in Eq. (19), we conclude that P_L^θ is the unique positive semi-definite solution to the discrete-time Lyapunov equation:

$$P_L^\theta = W_L + (K_s^\theta)^\top U_L K_s^\theta + \gamma_L \tilde{A}_\theta^\top P_L^\theta \tilde{A}_\theta.$$

The scalar constant v_L^θ is subsequently obtained as:

$$v_L^\theta = \frac{1}{1-\gamma_L} \left(\mathrm{Tr}(U_L \Sigma_F^\theta) + c_L(\theta) + \gamma_L(\mathrm{Tr}(P_L^\theta B \Sigma_F^\theta B^\top) + \mathrm{Tr}(P_L^\theta \Sigma_w)) \right).$$

Thus, the leader's value functions are fully determined analytically without requiring neural network approximations.

C.4.1 Easy Setting

Task Description The task is defined as an n-zone building thermal control problem with $n = 4$. The state $s_t \in \mathbb{R}^4$ represents temperature deviations, and the follower's action $b_t \in \mathbb{R}^2$ represents HVAC actuation. As defined previously, the system dynamics follow $s_{t+1} = A_\theta s_t + Bb_t + w_t$ with $w_t \sim \mathcal{N}(\mathbf{0}, \Sigma_w)$, and the leader's immediate reward is $r_L(s_t, b_t) = -(s_t^\top W_L s_t + b_t^\top U_L b_t + c_L(\theta))$.

For the leader's immediate reward, the specific weighting matrices W_L, U_L, and the configuration cost $c_L(\theta)$ are defined as:

$$W_L = \frac{1}{4}\left(I_4 - \frac{1}{4}\mathbf{1}\mathbf{1}^\top\right), \quad U_L = 0.5 \cdot I_2, \quad c_L(\theta) = 0.1 \cdot \left(\|\alpha\|_2^2 + \|\nu\|_2^2\right),$$

where I_n is the identity matrix, and $\mathbf{1} \in \mathbb{R}^n$ is a vector of ones.

The configurable state transition matrix $A_\theta \in \mathbb{R}^{4\times 4}$ is parameterized by the leader's decision variables, insulation levels $\alpha \in [0,1]^4$ and airflow levels $\nu \in [0,1]^4$, as follows:

$$A_\theta = \begin{bmatrix} 1+\Delta(\alpha_1) - h_{12}\nu_1 - h_{41}\nu_4 & h_{12}\nu_1 & 0 & h_{41}\nu_4 \\ h_{21}\nu_1 & 1+\Delta(\alpha_2) - h_{23}\nu_2 - h_{21}\nu_1 & h_{23}\nu_2 & 0 \\ 0 & h_{32}\nu_2 & 1+\Delta(\alpha_3) - h_{34}\nu_3 - h_{32}\nu_2 & h_{34}\nu_3 \\ h_{41}\nu_4 & 0 & h_{43}\nu_3 & 1+\Delta(\alpha_4) - h_{41}\nu_4 - h_{43}\nu_3 \end{bmatrix}$$

where $\Delta(\alpha_i) := k_i(1-\alpha_i)$. The positive coefficients k_i and h_{ij} representing specific physical constants are set as:

$$\begin{aligned} k_1 &= 0.04, \quad h_{12} = 0.05, \quad h_{41} = 0.05 \\ k_2 &= 0.03, \quad h_{23} = 0.04, \quad h_{21} = 0.03 \\ k_3 &= 0.06, \quad h_{34} = 0.06, \quad h_{32} = 0.04 \\ k_4 &= 0.05, \quad h_{41} = 0.05, \quad h_{43} = 0.03 \end{aligned}$$

The follower's immediate reward is defined as a quadratic cost:

$$r_F(s,b) = -s^\top W_F s - b^\top U_F b.$$

The specific weighting matrices for the follower are given by:

$$W_F = \begin{bmatrix} 8 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 6 \end{bmatrix}, \quad U_F = 0.01 \cdot I_2.$$

The control input matrix B and the unobservable disturbance covariance Σ_w are defined as:

$$B = \begin{bmatrix} 0.10 & 0 \\ 0.6 & 0 \\ 0 & 0.55 \\ 0 & 0.30 \end{bmatrix}, \quad \Sigma_w = 0.02 \cdot I_4.$$

Implementation Details In Line 7 of Algorithm 1, since the transition dynamics are linear-Gaussian and the reward functions are quadratic, we obtain $Q_F^{\theta\dagger}$ directly from the Discrete Algebraic Riccati Equation (DARE) solution for the follower's policy. For the same reason, in Line 9 of Algorithm 1, we analytically compute $Q_L^{\theta\dagger}$ and $V_L^{\theta\dagger}$ by solving the DARE. In Line 13 of Algorithm 1, we obtain $\nabla Q_F^{\theta\dagger}$ via the discrete Lyapunov equation. Finally, in Line 15 of Algorithm 1, the HG components are computed analytically using the known transition dynamics and policies, which bypasses the need for empirical trajectory sampling in these steps.

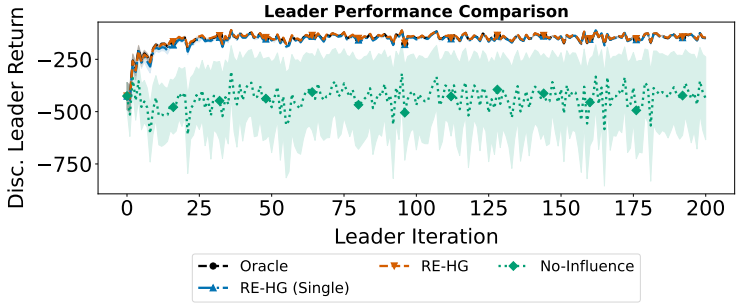


Figure 8: Results on Building Thermal Control. For each run, performance is evaluated as the mean of 50 test rollouts. Avg. (line) and std. (shaded area) across the 3 independent training runs are displayed.

Result Figure 8 presents the performance comparison in the Building Thermal Control task. In our setup, the initial state is stochastic, and the leader's initial parameters θ are sampled from a uniform distribution $\mathcal{U}(0, 0.3)$. To ensure the statistical reliability of the results despite this randomness, we evaluated the methods across three different environments, each with ten independent trials using distinct random seeds.

The results clearly demonstrate that RE-HG significantly outperforms NO-INFLUENCE. Notably, the performance of RE-HG is comparable to that of ORACLE. This indicates that our approach effectively recovers the necessary information to achieve near-optimal leader strategies.

Hyperparameters Hyperparameters used in this experiment are summarized in Table 5.

Table 5: Hyperparameters

Name	Symbol	Value
Leader iterations	K	200
Episodes per iteration	-	100
Evaluation episodes	-	50
Leader discount factor	γ_L	0.9
Follower discount factor	γ_F	0.9
Leader learning rate	α_L	0.1
Temperature	β	0.1
Gradient clip max norm	-	1.0
Local IRL iterations	-	200
Local IRL learning rate	-	0.01
threshold	ϵ	10^{-4}

C.4.2 Hard Setting (8-Zone Central Corridor Topology)

Task Description The task is extended to an n-zone building thermal control problem where $n = 8$. The state $s_t \in \mathbb{R}^8$ represents temperature deviations, and the follower's action $b_t \in \mathbb{R}^4$ represents HVAC actuation for the directly controlled zones. As defined previously, the system dynamics follow $s_{t+1} = A_\theta s_t + B b_t + w_t$ with $w_t \sim \mathcal{N}(\mathbf{0}, \Sigma_w)$, and the leader's immediate reward is $r_L(s_t, b_t) = -(s_t^\top W_L s_t + b_t^\top U_L b_t + c_L(\theta))$.

For the leader's immediate reward, the specific weighting matrices W_L, U_L, and the configuration cost $c_L(\theta)$ are defined as:

$$W_L = \frac{1}{8}\left(I_8 - \frac{1}{8}\mathbf{1}\mathbf{1}^\top\right), \quad U_L = 0.5 \cdot I_4, \quad c_L(\theta) = \|\alpha\|_2^2 + \|\nu\|_2^2,$$

where I_n is the identity matrix, and $\mathbf{1} \in \mathbb{R}^n$ is a vector of ones.

The configurable state transition matrix $A_\theta \in \mathbb{R}^{8\times 8}$ is parameterized by the leader's decision variables, insulation levels $\alpha \in [0,1]^8$ and airflow levels $\nu \in [0,1]^8$. The topology consists of two core corridors (Zones 1 and 2) and six perimeter offices (Zones 3–8). The off-diagonal entries $A_{i,j}$ (representing heat transfer from zone j to zone i) are:

$$A_{i,j} = \begin{cases} h_{cc}\nu_1 & \text{if } (i,j) \in \{(1,2),(2,1)\} \\ h_{cr}\nu_k & \text{if } i \in \{1,2\} \text{ and } j \text{ is an adjacent office} \\ h_{rc}\nu_k & \text{if } j \in \{1,2\} \text{ and } i \text{ is an adjacent office} \\ 0 & \text{otherwise} \end{cases}$$

where ν_k corresponds to the specific airflow parameter for the edge. Specifically, the edges connected to the West corridor (Zone 1) are controlled by ν_3 (Zone 3), ν_4 (Zone 4), and ν_6 (Zone 6). The edges connected to the East corridor (Zone 2) are controlled by ν_5 (Zone 5), ν_7 (Zone 7), and ν_8 (Zone 8). The parameter ν_2 is kept redundant to maintain the unified dimensionality $\theta \in [0,1]^{16}$. The diagonal entries are defined to preserve energy balance:

$$A_{i,i} = 1 + \Delta(\alpha_i) - \sum_{j \neq i} A_{j,i}$$

where $\Delta(\alpha_i) := k_i(1-\alpha_i)$. The physical heat transfer coefficients are set to $h_{cc} = 0.08$, $h_{cr} = 0.05$, $h_{rc} = 0.03$, and the insulation effects k_i are:

$$k_1 = 0.01, \quad k_2 = 0.01, \quad k_3 = 0.06, \quad k_4 = 0.08$$
$$k_5 = 0.06, \quad k_6 = 0.03, \quad k_7 = 0.04, \quad k_8 = 0.03$$

The follower's immediate reward is defined as a quadratic cost:

$$r_F(s,b) = -s^\top W_F s - b^\top U_F b.$$

The specific weighting matrices for the follower are given by:

$$W_F = \text{diag}(0.1, 0.1, 8.0, 30.0, 8.0, 3.0, 5.0, 3.0), \quad U_F = 0.01 \cdot I_4.$$

Note that Zone 4 (VIP office) is assigned the highest priority weight ($W_{F,44} = 30.0$) despite having no direct HVAC control, creating a challenging under-actuated control task for the follower.

The control input matrix $B \in \mathbb{R}^{8\times 4}$ and the unobservable disturbance covariance Σ_w are defined as:

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8 \end{bmatrix}, \quad \Sigma_w = 0.02 \cdot I_8.$$

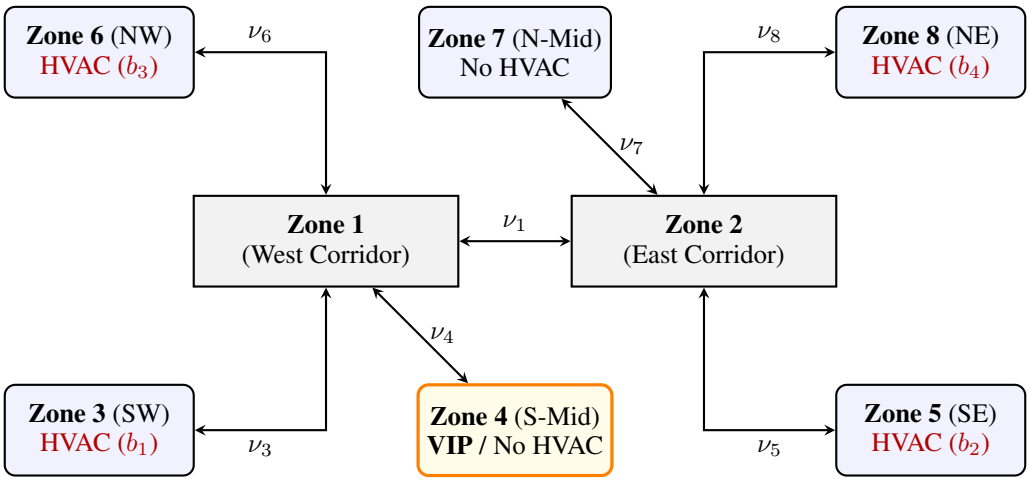
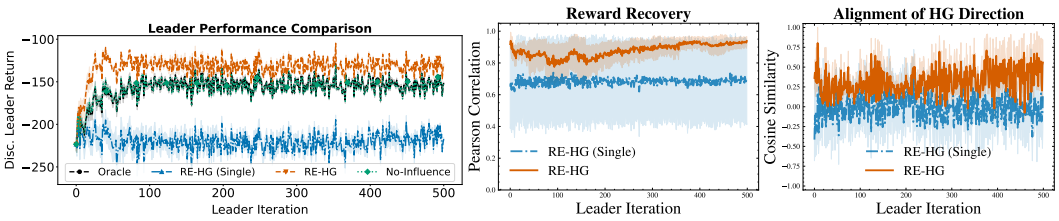


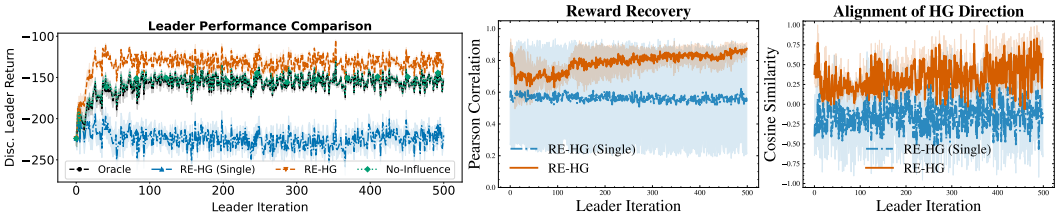
Figure 9: Topology of the 8-Zone Building Thermal Control Environment. Nodes represent the zones, with their respective HVAC control inputs. Edges represent the heat transfer between adjacent zones, parameterized by the leader's airflow configuration ν_k. Note that the VIP office (Zone 4) has no direct HVAC, highlighting the under-actuated nature of the follower's control task.

Implementation Details In Line 7 of Algorithm 1, since the transition dynamics are linear-Gaussian and the reward functions are quadratic, we obtain $Q_F^{\theta\dagger}$ directly from the DARE solution for the follower's policy. For the same reason, in Line 9 of Algorithm 1, we analytically compute $Q_L^{\theta\dagger}$ and $V_L^{\theta\dagger}$ by solving the DARE. At line 15 in Algorithm 1, we adopt the variance reduction baseline as formulated in (14). We utilize the relation $\nabla_\theta Q_F^{\theta\dagger} - \nabla_\theta V_F^{\theta\dagger} = \nabla_\theta \log g^{\theta\dagger}$, which is computed by the discrete Lyapunov equation. Furthermore, we use the expectation $\mathbb{E}[V_L^{\theta\dagger}(s_{t+1})]$ obtained via DARE as a baseline for the $\nabla_\theta \log p^\theta$ term to stabilize the stochastic gradient estimates. Finally, in Line 15 of Algorithm 1, the HG components are computed analytically using the known transition dynamics and policies, which bypasses the need for empirical trajectory sampling in these steps. To ensure stable gradient-based optimization, we apply gradient norm clipping with a maximum norm of 1.0. If the L2-norm of the gradient exceeds this threshold, we rescale it to maintain numerical stability during policy updates.



(a) Leader's discounted return (b) Reward Correlation (c) HG CosineSimilarity

Figure 10: Results for the relaxed setting ($W_{F,44} = 10.0$). (a) Leader's discounted return. For each run, performance is evaluated as the mean of 50 test rollouts. Avg. (line) and std. (shaded area) across the 5 independent training runs are displayed. Solid lines and shaded areas denote the avg. and std., respectively. (b) Pearson correlation between the recovered and true rewards, weighted by the undiscounted state-action occupancy measure of the surrogate follower policy. (c) Cosine similarity of the HG between the estimated and ORACLE



(a) Leader's discounted return (b) Reward Correlation (c) HG CosineSimilarity

Figure 11: Results for the relaxed setting ($W_{F,44} = 20.0$). (a) Leader's discounted return. For each run, performance is evaluated as the mean of 50 test rollouts. Avg. (line) and std. (shaded area) across the 5 independent training runs are displayed. Solid lines and shaded areas denote the avg. and std., respectively. (b) Pearson correlation between the recovered and true rewards, weighted by the undiscounted state-action occupancy measure of the surrogate follower policy. (c) Cosine similarity of the HG between the estimated and ORACLE.

Result As discussed in the ablation study in Section 5, we investigate the impact of the follower's reward landscape steepness on the optimization process. In the original Hard Setting, the highly skewed penalty for the VIP zone ($W_{F,44} = 30.0$) caused gradient explosions, trapping the ORACLE in a suboptimal local minimum.

Figure 10 and Figure 11 present the results when this steep penalty is relaxed to $W_{F,44} = 10.0$ and $W_{F,44} = 20.0$, respectively. As shown in the Figures 10 and 11, relaxing the reward steepness mitigates the gradient explosion. Consequently, the performance of the ORACLE significantly improves compared to the original $W_{F,44} = 30.0$ case, successfully finding superior leader policies. Meanwhile, RE-HG maintains robust and highly stable performance across all variations, consistently achieving high cosine similarity and optimal returns. This confirms that while the ORACLE is highly sensitive to sharp reward landscapes, the implicit smoothing mechanism of RE-HG inherently generates a manageable gradient landscape, reliably extracting effective HG regardless of the underlying reward steepness.

Hyperparameters Hyperparameters used in this experiment are summarized in Table 6.

C.5 2-player MG (Continuous)

Task Description The environment is modeled as a discretized linear system with a time step $\Delta t = 0.1$. The state $s_t = [p_t, v_t]^\top \in \mathbb{R}^2$ represents the position and velocity, and its evolution is governed by $s_{t+1} = As_t + Bb_t + Ca_t$, where a_t and b_t are the control inputs of the leader and the follower, respectively. The system matrices and the initial state s_0 are defined as:

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \Delta t \end{bmatrix}, \quad C = \begin{bmatrix} \Delta t \\ 0 \end{bmatrix}, \quad s_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Table 6: Hyperparameters

Name	Symbol	Value
Leader iterations	K	500
Episodes per iteration	-	100
Evaluation episodes	-	50
Leader discount factor	γ_L	0.9
Follower discount factor	γ_F	0.9
Leader learning rate	α_L	0.1
Temperature	β	0.1
Gradient clip max norm	-	1.0
Local IRL iterations	-	200
Local IRL learning rate	-	0.01
threshold	ϵ	10^{-2}

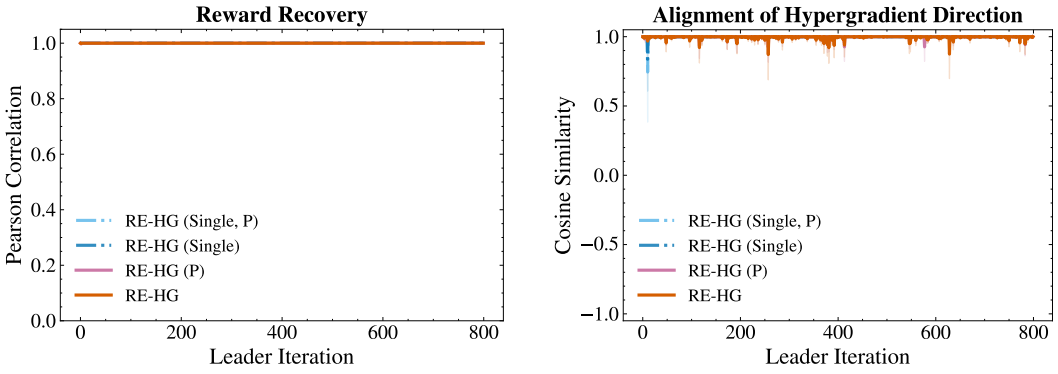
Reward Functions The follower maximizes the quadratic reward $r_F(s_t, b_t) = -s_t^\top W_F s_t - b_t^\top U_F b_t$ to stabilize at the origin, with $W_F = I_2$ and $U_F = [1]$. The leader aims to induce oscillation around the target $s_L^\star = [-0.8, 0]^\top$ by maximizing a Gaussian kernel reward:

$$r_L(s_t) = \exp\left(-\frac{1}{2}(s_t - s_L^\star)^\top M (s_t - s_L^\star)\right),$$

where $M = 0.25 I_2$. Note that while r_L depends only on the state s_t, it is implicitly coupled with the actions a_t and b_t through the system dynamics.

Implementation Details Before the main training loop, the leader's critic Q_L is pre-trained using 40 episodes generated from a randomized initial distribution where $s_0[0] \sim \mathcal{U}(-1.5, 1.5)$ and $s_0[1] \sim \mathcal{U}(-1.0, 1.0)$. This dataset comprises approximately 4,000 transitions based on a 100-step limit per episode, which are used to perform 50 optimization epochs via SARSA-TD updates to stabilize the critic initialization. In Line 5 of Algorithm 1, if the transition dynamics are unknown, we estimate them via linear regression. In Line 13 of Algorithm 1, we obtain $\nabla Q_F^{\theta\dagger}$ via the discrete Lyapunov equation. In Line 12 of Algorithm 1, the benefit B_L is calculated as the difference between the online leader's action-value function (Q_L^{online}) and the target leader's value function (V_L^{target}). To ensure stable gradient-based optimization, we apply gradient norm clipping with a maximum norm of 1.0. If the L2-norm of the gradient exceeds this threshold, we rescale it to maintain numerical stability during policy updates.

Result As discussed in Section 5, we present the detailed evaluation metrics for the continuous 2-player Markov Game task. Figure 12a shows the highly stable weighted Pearson correlation between the recovered and true rewards. Figure 12b illustrates the cosine similarity of the hypergradient direction against ORACLE. As shown in Figure 13, calculating the benefit using the online network for both functions (i.e., $Q_L^{\text{online}} - V_L^{\text{online}}$) fails to achieve stability. This instability arises because, near the optimal policy, the influence term practically vanishes. While Q_L learns and generalizes at this point, the leader's policy continues to change due to other gradient terms. Consequently, the previously generalized Q_L values become inaccurate for the shifted policy, leading to performance degradation.



(a) Reward Correlation

(b) HG Cosine Similarity

Figure 12: (a) Pearson correlation between the recovered and true rewards, weighted by the undiscounted state-action occupancy measure of the surrogate follower policy. (b) Cosine similarity of the HG between the estimated and ORACLE.

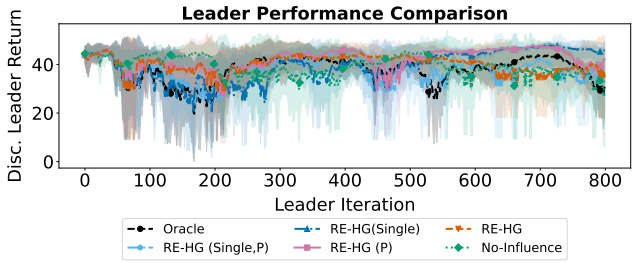


Figure 13: This figure illustrates the instability caused by calculating the benefit using the online network for both functions. Specifically, the benefit is calculated as the difference between the online leader's action-value function (Q_L^{online}) and the online leader's value function (V_L^{online}). For each run, performance is evaluated as the mean of 10 test rollouts. Avg. (line) and std. (shaded area) across the 3 independent training runs are displayed. "P" indicates the known transition probability scenario.

Hyperparameters Hyperparameters used in this experiment are summarized in Table 7.

Table 7: Hyperparameters

Name	**Symbol**	**Value**
Time step	Δt	0.1
Max steps per episode	T	100
Leader iterations	K	800
Episodes per iteration	-	4
Leader discount factor	γ_L	0.99
Follower discount factor	γ_F	0.99
Leader learning rate	α_L	0.05
Temperature	β	0.01
Leader Critic Update times	$-$	5
Leader sigma	Σ_L	1.0×10^{-3}
threshold	ϵ	10^{-4}
Pre-training episodes	n_{pre}	40
Pre-training epochs	-	50

C.6 Comparing Oracle

In the Hard Setting of Section C.4.2, RE-HG attains a higher leader return than ORACLE. Since ORACLE computes the exact hypergradient, the gap may simply reflect the smaller effective update of RE-HG. We test this by retuning the ORACLE update in three ways: (a) a tuned learning rate, (b) gradient clipping, and (c) a scaled influence term. Modification (c) is the closest ORACLE-side analogue of the regularization in RE-HG, since the eigenvalue truncation in Algorithm 2 has no counterpart when the true reward is available.

All ablations use the continuous Conf-MDP environment with the same base configuration as the Hard Setting of Section C.4.2, and each ORACLE and RE-HG variant is evaluated over five random seeds. Figure 14 shows the learning curves.

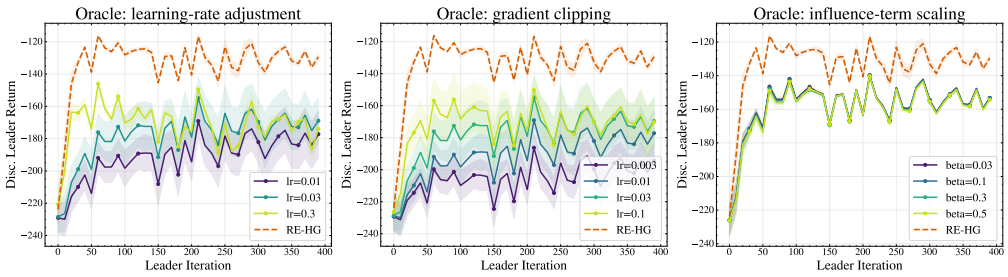


Figure 14: Learning curves for three ORACLE-side modifications. Lines represent the mean discounted leader return across five random seeds, and shaded regions represent one standard deviation. **(a) Tuned learning rate:** the ORACLE learning rate is varied. **(b) Gradient clipping:** the hypergradient is clipped and the learning rate is varied. **(c) Influence-term scaling:** the influence term is multiplied by a global factor while the direct term is left unchanged. The dashed RE-HG curve is included as the baseline in each panel.

Tuning the learning rate does not help. The ORACLE uses $\alpha_L = 0.1$ in the Hard Setting, and Figure 14(a) shows that none of the alternatives 0.01, 0.03, and 0.3 improved on it, the best being -170.54 ± 14.07. Clipping the hypergradient as $g \leftarrow g/\|g\|$ fares no better across learning rates 0.003, 0.01, 0.03, and 0.1, the last of which is the original value, with a best of -171.12 ± 14.73 in Figure 14(b).

Scaling the influence term, the second term of the hypergradient, by factors between 0.03 and 0.5 while leaving the direct term unchanged does help. Figure 14(c) shows it raises the ORACLE return to between -155.28 ± 0.72 and -154.15 ± 0.00, still far below the -129.59 ± 0.88 of RE-HG.

Rescaling the update therefore does not reproduce the effect of eigenvalue truncation. Modifications (a) and (b) change only the magnitude of the update, and (c) changes the ratio between the direct and influence terms by a single global factor. Eigenvalue truncation instead suppresses poorly excited directions of the influence term selectively, so it changes the direction of the update rather than its scale.

C.7 Multi-environment IRL

A natural alternative to the sequential aggregation of Algorithm 2 is to estimate one shared reward jointly from the demonstrations accumulated across all environments visited so far. We ask whether sequential aggregation gives anything up.

We implemented a joint multi-environment IRL baseline from Section 7.2 of Cao et al. (2021). That formulation assumes access to the transition dynamics, so we evaluated both methods in a discrete environment with known dynamics. Figure 15 shows the learning curves.

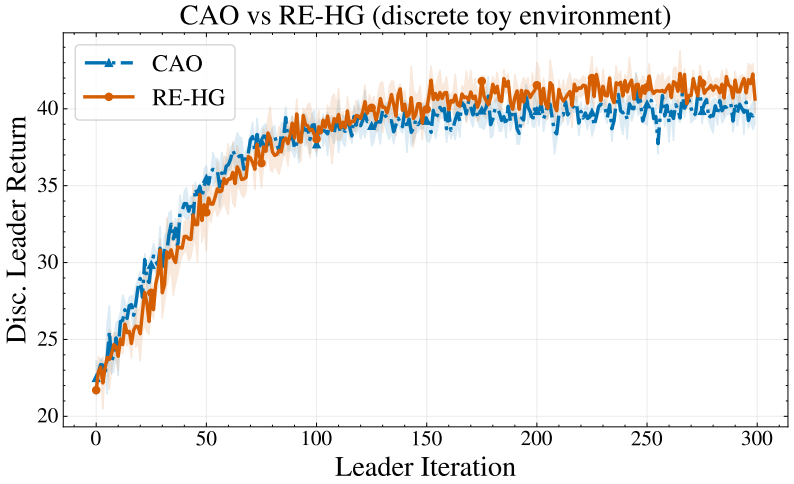


Figure 15: Comparison between the joint multi-environment IRL baseline (CAO) and RE-HG in the discrete toy environment with known transition dynamics. Lines represent the mean discounted leader return across three random seeds, and shaded regions represent one standard deviation.

The two methods match throughout training and both converge close to the optimum, reaching discounted leader returns of 21.46 and 20.75 respectively, averaged over the last 30 leader iterations with the same three random seeds. Sequential aggregation is therefore no better in final performance, which is what our analysis predicts: both approaches resolve reward ambiguity under their respective idealized assumptions. Its advantages are algorithmic, and are threefold.

First, RE-HG integrates reward estimation into the standard bilevel optimization procedure. Each leader update presents the follower with a new environment, and RE-HG re-estimates the reward from trajectories collected in that environment and immediately computes the hypergradient. Joint multi-environment IRL instead estimates a shared reward from demonstrations accumulated across environments, which requires storing past trajectories and deciding how many of them to keep.

Second, the formulation of Cao et al. (2021) assumes known transition dynamics, whereas RE-HG also applies to the two-player Markov game setting of this paper, where they are unknown.

Third, Theorem 1 and Proposition 2 are developed for sequential reward aggregation and give identifiability together with an error bound relating the reward-estimation error to the hypergradient error. We are not aware of a joint multi-environment IRL method that applies directly to our setting and offers a comparable guarantee.

C.8 Misspecified reward features

Assumption 1 assumes a known feature extractor. One-hot features make this harmless in discrete environments, but in continuous ones the recovered reward is only as good as the chosen basis, so we test three forms of misspecification: basis mismatch, limited update capacity, and redundant features. All experiments use the continuous Conf-MDP (LQR) environment with the same base configuration as the Hard Setting of Section C.4.2; each condition is evaluated over five random seeds. Figure 16 shows the learning curves.

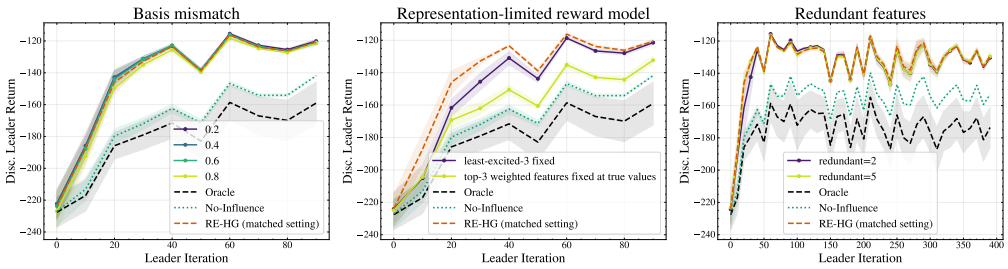


Figure 16: Learning curves under reward-feature misspecification in the continuous Conf-MDP. Lines show mean discounted leader return and shaded regions show one standard deviation. The misspecified RE-HG runs use three random seeds, while the Oracle, No-Influence, and matched-setting RE-HG reference curves use five seeds. **(a) Basis mismatch:** the true reward contains off-diagonal interaction terms such as $s_i s_j$, whereas the IRL model is restricted to diagonal quadratic features. **(b) Limited update capacity and representation misspecification:** the three state-cost features with the largest true weights, accounting for approximately 80.4% of the total state-cost weight, are fixed at their true values to isolate an update-capacity restriction. A complementary curve fixes the three least-excited features at incorrect values, introducing genuine representation misspecification. **(c) Redundant-feature over-specification:** two or five cross-term features are added to the IRL model even though their true coefficients are zero. The RE-HG reference curve is included in every panel.

Under basis mismatch, Figure 16(a) shows that the exact analytic ORACLE becomes unstable because its HG is unregularized, yielding leader returns between -171.38 ± 10.72 and -155.81 ± 0.00. RE-HG stays between -128.40 ± 1.68 and -129.45 ± 2.33, consistent with the smoothing effect of eigenvalue truncation.

Figure 16(b) separates a restriction on update capacity from genuine representation error. Freezing the three state-cost features with the largest true weights, which account for $46/(0.1 + 0.1 + 8 + 30 + 8 + 3 + 5 + 3) \approx 80.4\%$ of the total state-cost weight, at their true values introduces no representation error and isolates the effect of restricting which directions can be updated; RE-HG still reaches -141.43 ± 2.50, well above the NO-INFLUENCE baseline of -153.21. Freezing the three least-excited features at incorrect values instead, $[1, 1, 1]$ against true values $[8, 3, 8]$, costs almost nothing: RE-HG reaches -129.94 ± 1.94, because eigenvalue truncation already suppresses these directions.

With redundant features, Figure 16(c) shows that RE-HG again beats the NO-INFLUENCE baseline, reaching -131.84 ± 1.60 with two redundant features and -130.92 ± 0.99 with five.

C.9 Optimal follower assumption

Proposition 1 characterizes the IRL solution only when the follower plays its exact best response, and a real follower is at best approximately optimal. We therefore test two suboptimal followers: a myopic one that runs a limited number of Riccati iterations, $T \in \{12, 16, 20, 30\}$, and an under-reactive one whose feedback gain is scaled as $(1 - \alpha)K^*$ with $\alpha \in \{0.05, 0.1, 0.2, 0.35\}$. Smaller T and larger α mean larger deviations from the optimal follower.

Both experiments use the continuous Conf-MDP (LQR) environment with the same base configuration as the Hard Setting of Section C.4.2; each condition is evaluated over three random seeds and the RE-HG baseline over five. Figure 17 shows the learning curves and Figure 18 the total-variation (TV) distance between the suboptimal follower and the exact best response during training.

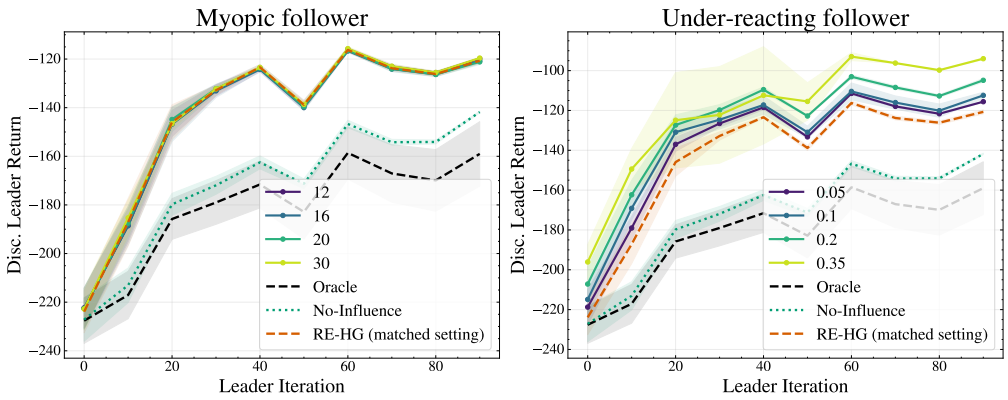


Figure 17: Learning curves under suboptimal follower models in the continuous Conf-MDP. Lines represent mean discounted leader return across three random seeds (five for the RE-HG baseline), and shaded regions represent one standard deviation. **(a) Myopic follower:** varying Riccati iteration budget T. **(b) Under-reactive follower:** varying gain reduction parameter α. The RE-HG curve is included as the baseline in each panel.

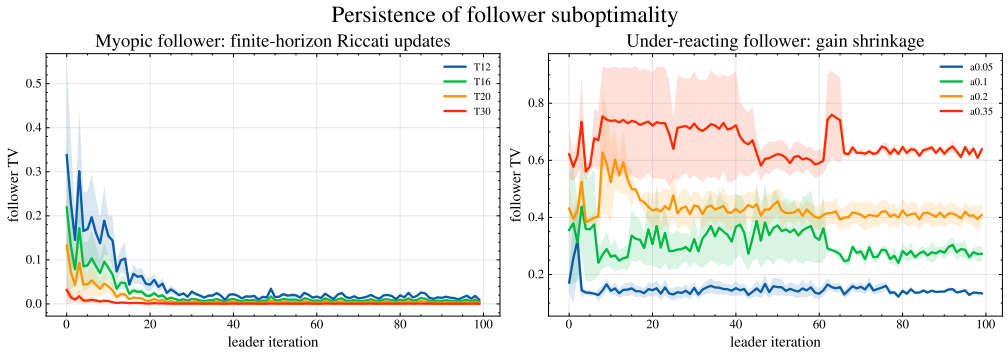


Figure 18: Total-variation distance between the suboptimal follower and the exact best response over the course of training, for the same two follower models and in the same panel order as Figure 17. **(a) Myopic follower. (b) Under-reactive follower.** Lines represent the mean across three random seeds, and shaded regions represent one standard deviation.

The myopic follower deviates only transiently. In Figure 18(a) the initial TV distance is about 0.34 for $T = 12$ but the average over training is 0.0443 ± 0.0153, and Figure 17(a) shows the leader return is essentially unaffected: RE-HG reaches -127.55 ± 0.46 for $T = 30$ and -127.91 ± 0.87 for $T = 12$.

The under-reactive follower deviates persistently, with TV distances between 0.15 and 0.66 throughout training in Figure 18(b). Here RE-HG adapts its reward estimate to the observed trajectories and, as Figure 17(b) shows, overtakes the ORACLE: -123.25 ± 0.42 for $\alpha = 0.05$ and -99.74 ± 0.17 for $\alpha = 0.35$, against -154.28 ± 2.35 to -152.36 ± 0.79 for the ORACLE across the tested under-reaction levels. The reward correlations stay between 0.70 and 0.82. The comparison is not symmetric, however: the ORACLE uses the true reward but still assumes an exact best response, which is violated here by construction.

Taken together, RE-HG tolerates both transient and persistent follower suboptimality by fitting its reward estimate to the behavior actually observed.

D Broader Impacts

This research can positively impact the efficiency of real-world hierarchical systems, such as smart grids and traffic control, where the follower's reward is unknown. However, if applied to human-

centric domains like recommendation systems, inferring unobserved rewards from user behavior could lead to unintended manipulation or privacy concerns. To mitigate these risks, such applications require transparency regarding data usage and strict safety constraints on the leader's action space.

E Compute Resources

All experiments were conducted on a workstation equipped with dual Intel(R) Xeon(R) Gold 6348 CPUs @ 2.60GHz (totaling 56 physical cores and 112 logical threads) and 512 GB of RAM. The entire suite of experimental evaluations, including all baseline comparisons and ablations, was completed within 12 hours of total wall-clock time.