Proceedings of the Thirty-Sixth International Conference on Automated Planning and Scheduling (ICAPS 2026)

Sample-Efficient Hypergradient Estimation for Decentralized Bi-Level Reinforcement Learning

Mikoto Kudo[1,2], **Takumi Tanabe**[3], **Akifumi Wachi**[3], **Youhei Akimoto**[1,2]
[1]University of Tsukuba,
[2]RIKEN AIP,
[3]LY Corporation
mikoto@bbo.cs.tsukuba.ac.jp, {akifumi.wachi, takumi.tanabe}@lycorp.co.jp, akimoto@cs.tsukuba.ac.jp

Abstract

Many strategic decision-making problems, such as environment design for warehouse robots, can be naturally formulated as bi-level reinforcement learning (RL), where a leader agent optimizes its objective while a follower solves a Markov decision process (MDP) conditioned on the leader's decisions. In many situations, a fundamental challenge arises when the leader cannot intervene in the follower's optimization process; it can only observe the optimization outcome. We address this decentralized setting by deriving the hypergradient of the leader's objective, i.e., the gradient of the leader's strategy that accounts for changes in the follower's optimal policy. Unlike prior hypergradient-based methods that require extensive data for repeated state visits or rely on gradient estimators whose complexity can increase substantially with the high-dimensional leader's decision space, we leverage the Boltzmann covariance trick to derive an alternative hypergradient formulation. This enables efficient hypergradient estimation solely from interaction samples, even when the leader's decision space is high-dimensional. Additionally, to our knowledge, this is the first method that enables hypergradient-based optimization for 2-player Markov games in decentralized settings. Experiments highlight the impact of hypergradient updates and demonstrate our method's effectiveness in both discrete and continuous state tasks.

Code — https://github.com/akimotolab/BC-HG

1 Introduction

Bi-level reinforcement learning (RL) is a hierarchical framework capable of representing complex objective structures. In bi-level RL, the lower-level (i.e., *follower*) typically involves policy optimization within a *Configurable MDP* (Metelli, Mutti, and Restelli 2018), which is an MDP parameterized by external variables $\theta \in \Theta$. Meanwhile, the objective function of the upper-level (i.e., *leader*) depends on the lower-level optimal policy, creating a nested dependency structure as follows:

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

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

where Θ is the set of possible parameters in the upper-level, and $\mathcal{G}$ is the set of all possible policies in the lower-level. The set of lower-level optimal policies, $\mathcal{G}^\dagger(\theta)$, depends on the upper-level variable and is called the *best response* to θ.

Due to its versatility in modeling hierarchical dependencies, bi-level RL has been applied to many domains, including reinforcement learning from human feedback (RLHF) (Chakraborty et al. 2023; Shen, Yang, and Chen 2024; Yang, Gao, and Yuan 2025), reward shaping (Shen, Yang, and Chen 2024; Yang, Gao, and Yuan 2025), safe RL (Zheng and Gu 2025), model-based RL (Rajeswaran, Mordatch, and Kumar 2020), and advertising planning (Muneeb et al. 2019). Yet an important challenge remains under-explored: decentralized learning, where the leader cannot intervene in the follower's optimization process and must treat the follower algorithm as fixed. This occurs when followers rely on pre-defined standard algorithms that are impractical or undesirable to modify. For example, in environment design for autonomous warehouse robots, the designer optimizes environment parameters while leaving each robot's built-in adaptation algorithm (e.g., LQR) unchanged. The technical challenge is then to estimate leader updates that account for best-response shifts without controlling follower optimization.

To address this challenge, we propose a method for decentralized bi-level RL, where the leader is agnostic to the follower's algorithm and cannot control follower updates. We study two settings: 1) configurable MDPs, where the leader influences the follower through environment/configuration parameters; and 2) Markov games (MGs), where both leader and follower have policies and influence each other through policy interaction. In both settings, the upper-level objective is the leader's discounted cumulative reward, while the lower-level objective is the follower's entropy-regularized discounted cumulative reward. For each setting, we derive the hypergradient of the upper-level objective, which can be estimated from online interaction samples consisting of states, follower and leader actions, and the leader's immediate rewards. By updating the leader's variables or policy using the estimated hypergradient, our method enables steepest ascent updates that "anticipate" changes in the follower's best response. We also assume access to the follower's actions (during exploration and exploitation) and learning outcomes. This information-access assumption makes online

---

hypergradient estimation tractable while preserving the decentralized, non-interventionist setting.

Two existing methods applicable to our configurable MDP setting, HPGD (Thoma et al. 2024) and SoBiRL (Yang, Gao, and Yuan 2025), require multiple visits to the same state within a sample batch to estimate the hypergradient. However, in continuous state spaces or in large discrete state spaces relative to batch size, this requirement is impractical without generating additional trajectories or arbitrary state resets, severely limiting real-world applicability. Our derived hypergradient circumvents this issue via the "Boltzmann covariance trick," enhancing scalability by enabling estimation solely from interaction samples, even in continuous-state tasks with high-dimensional configuration spaces. Furthermore, we extend this result to 2-player MGs. To our knowledge, this is the first work to derive the hypergradient for 2-player MGs under entropy regularization.

We empirically evaluate our proposed approach on Configurable MDPs and MGs with both discrete and continuous state spaces. In Configurable MDP settings, our method demonstrates robustness against weak entropy regularization and scalability to high-dimensional leader parameters. Prior methods like HPGD often degrade or become trapped in local minima due to estimation errors in these scenarios. Furthermore, in MG settings, our method identifies policies that successfully induce favorable follower behaviors, which the baselines fail to discover. These results validate that our method enables effective hypergradient-based optimization solely from interaction samples.

2 Preliminaries

Entropy Regularization Throughout this paper, we assume that a follower solves a Markov decision process (MDP) with entropy regularization. It is formulated as follows. Let $(\mathcal{S}, \mathcal{B}, p, \rho_0, r_F, \gamma_F)$ be the MDP, where $\mathcal{S}$ is the state space; $\mathcal{B}$ the action space of the follower; $p : \mathcal{S} \times \mathcal{B} \times \mathcal{S} \to [0, 1]$ the state transition probability; $\rho_0 : \mathcal{S} \to [0, 1]$ the initial state distribution; $r_F : \mathcal{S} \times \mathcal{B} \to \mathbb{R}$ the reward function of the follower; and $\gamma_F \in [0, 1)$ the discount factor for the follower. The objective of the follower is to find a stochastic policy $g \in \mathcal{G} : \mathcal{S} \times \mathcal{B} \to [0, 1]$ that maximizes the expected cumulative reward under entropy regularization:

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

where $\mathcal{G}$ is the set of all possible policies on $\mathcal{S} \times \mathcal{B}$; $\beta > 0$ is the constant determining the strength of the entropy regularization; $H(g; s) := -\mathbb{E}_{b\sim g(\cdot|s)}[\log g(b \mid s)]$ is the entropy of g given $s \in \mathcal{S}$; and the expectation is taken for a trajectory $\tau := (s_0, b_0, s_1, b_1, \dots)$ generated by g, i.e., $p(\tau \mid g) = \rho_0(s_0) \prod_{t=0}^{\infty} p(s_{t+1} \mid s_t, b_t) g(b_t \mid s_t)$.

The main technical reason to assume entropy regularization is to guarantee the existence and uniqueness of the optimal policy as a stochastic policy. Entropy regularization is often assumed in inverse RL algorithms as a natural approach to model the stochasticity in the expert policy. Under entropy regularization, the value and action value functions (also called the soft value and soft action value functions) are defined as

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

and $Q_F^g(s, b) := r_F(s, b) + \gamma_F \mathbb{E}_{s'\sim p(\cdot|s,b)}[V_F^g(s')]$. When the policy model is sufficiently expressive, the optimal policy is uniquely determined by the optimal value functions $V_F^\star(s) = \max_{g\in\mathcal{G}} V_F^g(s)$ and $Q_F^\star(s, b) = \max_{g\in\mathcal{G}} Q_F^g(s, b)$ for each $s \in \mathcal{S}$ as a Boltzmann distribution $g^\star(b \mid s) = \exp\left(\beta^{-1}\left(Q_F^\star(s, b) - V_F^\star(s)\right)\right)$. The uniqueness of the optimal policy is essential for bi-level RL problems.

Configurable Markov Decision Process A configurable Markov decision process (configurable MDP) generalizes the standard MDP by introducing a configurable parameter that parameterizes the state transition probability and the reward function. A configurable MDP $\mathcal{M}_\theta$, parameterized by $\theta \in \Theta$, is defined by the tuple $(\mathcal{S}, \mathcal{B}, p^\theta, \rho_0^\theta, r_F^\theta, \gamma_F)$, where, differently from the standard MDP, p^θ, ρ_0^θ, and r_F^θ are parameterized by $\theta \in \Theta$. Given $\mathcal{M}_\theta$ for a fixed $\theta \in \Theta$, the follower maximizes the expected cumulative reward under entropy regularization; that is, it solves (1) with the parameterized objective J_θ, where p, ρ_0, and r_F are replaced by their parameterized counterparts p^θ, ρ_0^θ, and r_F^θ.

The leader's objective is to maximize its utility given the follower's best response. This is formulated as a bi-level reinforcement learning problem:

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

where J_L denotes the leader's utility under configuration θ, and $g^{\theta\dagger}$ is the follower's best response to θ. Due to entropy regularization in the lower-level problem, $g^{\theta\dagger}$ is uniquely determined for every $\theta \in \Theta$ and corresponds to the θ-conditioned optimal 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 value functions of $\mathcal{M}_\theta$. Consequently, this problem is well-defined due to the existence and uniqueness of the best response $g^{\theta\dagger}$. In the subsequent sections, we assume that the follower's policy always coincides with the optimal policy specified above. This assumption is not overly restrictive, as non-asymptotic convergence to an ϵ-optimal policy for the follower has been established by Thoma et al. (2024) for standard entropy-regularized RL algorithms, such as Soft Value Iteration, Q-learning, and Natural Policy Gradient.

In this paper, we focus on the scenario where the leader's utility is defined by the discounted cumulative reward, which depends on the follower's policy, and a regularization term that depends only on the configuration:

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

where γ_L is the leader's discount factor, r_L^θ is the leader's reward function, and Φ_L defines the regularization for configuration θ. Here, Φ_L is a task-specific regularizer for the leader (e.g., an ℓ_2 penalty on θ), and is distinct from the entropy regularization used in the follower objective.

2-Player Markov Game A 2-player MG can be viewed as an extension of configurable MDPs, where the leader acts as an agent with a policy parameterized by $\theta \in \Theta$, and the environment is influenced indirectly through the leader's actions. It is defined by the tuple ($\mathcal{S}$, $(\mathcal{A}, \mathcal{B})$, p, ρ_0, (r_L, r_F), (γ_L, γ_F)), where $\mathcal{S}$ is the state space; $\mathcal{A}$ and $\mathcal{B}$ the action spaces of the leader and the follower, respectively; $p : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times \mathcal{S} \to [0, 1]$ the state transition probability; $\rho_0 : \mathcal{S} \to [0, 1]$ the initial state distribution; r_L and $r_F : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ the reward functions of the leader and the follower; and $\gamma_L, \gamma_F \in [0, 1)$ the discount factors for the leader and the follower. The leader's policy and the follower's policy are denoted as $f_\theta : \mathcal{S} \times \mathcal{A} \to [0, 1]$ and $g : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to [0, 1]$, respectively.[1] Let the leader's policies be parameterized by $\theta \in \Theta$. Similar to the configurable MDP case, the follower maximizes the expected cumulative reward under entropy regularization, i.e., it solves (1). The difference lies in that $r_F^\theta(s, b)$ is replaced by $r_F(s, a, b)$ and $p^\theta(s' \mid s, b)$ is replaced by $p(s' \mid s, a, b) f_\theta(a' \mid s')$. That is, instead of directly controlling the follower's reward and state transition, these are indirectly affected by the leader's action. The leader's optimization problem is identical to (2), where the expected cumulative reward J_L is defined as:

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

Analogous to the configurable MDP case, we assume that the follower's policy is optimal.

We define the value and action-value functions for the follower g under the leader 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 $Q_F^{\theta g}(s, a, b) := r_F(s, a, b) + \gamma_F \mathbb{E}_{s'} \mathbb{E}_{a'}\left[V_F^{\theta g}(s', a')\right]$, where the expectation $\mathbb{E}_\tau$ is taken over a trajectory generated by f_θ and g, i.e., $p(\tau \mid \theta, g) = \rho_0(s_0) \prod_{t=0}^{\infty} p(s_{t+1} \mid s_t, a_t, b_t) g(b_t \mid s_t, a_t) f_\theta(a_t \mid s_t)$, $\mathbb{E}_{s'}$ is taken over next state $s' \sim p(\cdot \mid s, a, b)$, and $\mathbb{E}_{a'}$ is taken over next leader action $a' \sim f_\theta(\cdot \mid s')$. In the definition of $V_F^{\theta g}$, the leader's action a is given as an input because it is observed before the follower's decision.

The leader's action-value functions are defined as:

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

Note that the leader does not employ entropy regularization; hence, the value functions for the leader are the standard ones. For conciseness, the leader's action-value function under the leader policy f_θ and the corresponding best response $g^{\theta\dagger}$ is denoted as $Q_L^{\theta\dagger} := Q_L^{\theta g^{\theta\dagger}}$. In addition, the follower's optimal value and action-value functions under f_θ, i.e., the value functions of f_θ and the optimal policy $g^{\theta\dagger}$, are denoted as $V_F^{\theta\dagger} := V_F^{\theta g^{\theta\dagger}}$ and $Q_F^{\theta\dagger} := Q_F^{\theta g^{\theta\dagger}}$.

3 Related Works

Bi-Level RL with Cumulative Upper-Level Objectives in Configurable MDPs Several studies address bi-level RL with cumulative reward objectives at the upper level, such as Thoma et al. (2024) and Yang, Gao, and Yuan (2025). Both derive hypergradients assuming access to the follower's entropy-regularized ϵ-optimal best response, similar to our setting. Consequently, their methods are applicable to decentralized follower scenarios.

However, these approaches share a common limitation: hypergradient estimation requires multiple trajectories originating from the same initial state with diverse initial follower actions. Such trajectories are generally inaccessible unless the state space is discrete and small enough to allow repeated visits to identical states. Thoma et al. (2024) circumvent this issue by assuming access to an oracle capable of generating trajectories from arbitrary initial states and actions. Although this challenge can be mitigated without such an oracle, either by constraining the dimensionality of upper-level variables as in Thoma et al. (2024) or by truncating the hypergradient to its first-step component as in Yang, Gao, and Yuan (2025), these simplifications and approximations degrade performance in practical settings as seen later.

In contrast, we derive an alternative hypergradient formulation that can be estimated solely from experienced trajectories. Our method eliminates the need for additional trajectories, enabling scalability to large state spaces and high-dimensional leader parameters.

Bi-Level RL in Markov Games In 2-player MG settings with a decentralized follower, three main approaches have been proposed: two-timescale gradient updates (Rajeswaran, Mordatch, and Kumar 2020; Vu et al. 2022), total derivative gradient approaches (Yang et al. 2023), and operator-based methods (Zhang et al. 2020). Two-timescale gradient updates (Rajeswaran, Mordatch, and Kumar 2020; Vu et al. 2022) employ the first-order gradient for the leader's updates and two different update frequencies to maintain the follower's optimality. However, this approach may fail to find the optimal leader policy, particularly when the leader's and follower's objectives conflict, as demonstrated in Section 5.3. Yang et al. (2023) propose a total-derivative gradient method, ST-MADDPG. However, they assume a deterministic follower policy unconditioned on the leader's action. This, combined with the lack of mathematical detail regarding policy updates, makes adapting their method to our setting nontrivial. Moreover, this method requires computing the inverse Hessian product with respect to policy parameters, often incurring substantial computational

---

[1] In this paper, we consider the case where the follower observes the leader's action before making a decision. However, it is straightforward to extend our results to cases where the follower does not observe the leader's action, i.e., $g : \mathcal{S} \times \mathcal{B} \to [0, 1]$.

costs and resulting in unstable updates. Bi-AC (Zhang et al. 2020), on the other hand, is applicable with minor adjustments. This method builds on the Stackelberg stage-game operator, similar to Nash-Q (Hu and Wellman 2003). However, this operator is not guaranteed to converge to the optimal leader policy or even to improve the leader's performance. Further discussion is provided in Section 5.3.

Our proposed method leverages the hypergradient to account for changes in the follower's best response during the leader's update, similar to total derivative gradient approaches (Yang et al. 2023), but without the computationally expensive inverse Hessian product.

4 Proposed Approach

A key difference from prior studies (Thoma et al. 2024; Yang, Gao, and Yuan 2025) is that we leverage the *Boltzmann covariance trick* to resolve their limitation of requiring expectations over follower actions from the same state for hypergradient estimation. This constraint necessitates elaborate trajectory collection procedures, which restricts its applicability. Our method avoids this computation and enables hypergradient estimation from limited interaction history. To our knowledge, this is the first work deriving the hypergradient for 2-player MGs under entropy regularization.

Hypergradient for Configurable MDPs The hypergradient of J_L is the hypergradient of the cumulative reward (first term in (3)) plus the gradient of the regularization term (second term), $\nabla\Phi_L(\theta)$, which is assumed to be analytically differentiable. Consequently, we omit the regularization term in the following derivation for simplicity.

To derive the hypergradient, we define the leader's value function under the follower's action as

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

Unlike the standard action-value function, the inputs are the state s and the follower's action b, rather than the leader's action. Let $Q_L^{\theta\dagger} := Q_L^{\theta g^{\theta\dagger}}$ be the leader's value function under the best response $g^{\theta\dagger}$. Then, we define the *Benefit* of the leader when the follower takes an action b at state s as

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

A positive Benefit indicates that the follower's action is relatively beneficial to the leader. Thus, the leader aims to encourage the follower to select actions with a positive Benefit more frequently.

Using the Benefit, we can compute the hypergradient as follows. Its proof is provided in Appendix B.2.

Theorem 1. *Suppose the gradients $\nabla_\theta r_L^\theta$, $\nabla_\theta r_F^\theta$, $\nabla_\theta \log \rho_0^\theta$, and $\nabla_\theta \log p^\theta$ are computable everywhere. The hypergradient of* (3) *is given by*

$$\begin{aligned}\nabla_\theta J_L(\theta, g^{\theta\dagger}) = \mathbb{E}_\tau\Bigg[\sum_{t=0}^{\infty}\gamma_L^t\bigg(&\nabla_\theta r_L^\theta(s_t,b_t)\\ &+ V_L^{\theta\dagger}(s_t)\nabla_\theta \log p^\theta(s_t \mid s_{t-1}, b_{t-1})\\ &+ \frac{B_L^{\theta\dagger}(s_t,b_t)}{\beta}\nabla_\theta Q_F^{\theta\dagger}(s_t,b_t)\bigg)\Bigg], \end{aligned} \quad (5)$$

where $p^\theta(s_0 \mid s_{-1}, b_{-1})$ refers to $\rho_0^\theta(s_0)$, and

$$\begin{aligned}\nabla_\theta Q_F^{\theta\dagger}(s,b) = \mathbb{E}_\tau\Bigg[&\sum_{t=0}^{\infty}\gamma_F^t\nabla_\theta r_F^\theta(s_t,b_t)\\ &+ \gamma_F^{t+1}V_F^{\theta\dagger}(s_{t+1})\nabla_\theta \log p^\theta(s_{t+1} \mid s_t, b_t)\ \Bigg|\ \begin{matrix} s_0 = s \\ b_0 = b \end{matrix}\Bigg]. \end{aligned} \quad (6)$$

The first two terms of (5) are interpreted as the direct effect of the differentiation of the cumulative reward resulting from the differentiation of the reward function and the transition probability. The third term is interpreted as the indirect effect due to the change in the follower's best response.

Thoma et al. (2024) derive a hypergradient for the same bi-level RL problem, albeit with a different formulation. Importantly, since our hypergradient formulation is mathematically equivalent to that of Thoma et al. (2024), the non-asymptotic convergence guarantees established in their work apply directly to our method. Our contribution is thus complementary: we retain the same gradient target but introduce a more efficient and oracle-free estimation mechanism, making the approach applicable in practical decentralized settings.

More specifically, the key distinction lies in the form of the third term of the hypergradient, leading to different implementations of hypergradient estimation. The third term of the hypergradient in (Thoma et al. 2024) is derived as

$$\frac{1}{\beta}\mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma_L^t Q_L^{\theta\dagger}(s_t,b_t)\nabla_\theta A_F^{\theta\dagger}(s_t,b_t)\right], \quad (7)$$

where $A_F^{\theta\dagger}(s,b) := Q_F^{\theta\dagger}(s,b) - V_F^{\theta\dagger}(s)$. This requires estimating the gradients $\nabla_\theta Q_F^{\theta\dagger}(s,b)$ derived in (6) and $\nabla_\theta V_F^{\theta\dagger}(s) = \mathbb{E}_{b\sim g^{\theta\dagger}(\cdot|s)}\left[\nabla_\theta Q_F^{\theta\dagger}(s,b)\right]$ (derived in the proof). However, there is a difficulty in estimating these two gradients from samples. These gradients are estimated by Monte Carlo using trajectories of the follower's policy. The estimation of $\nabla_\theta Q_F^{\theta\dagger}(s,b)$ requires trajectories starting at state s and follower action b. On the other hand, the estimation of $\nabla_\theta V_F^{\theta\dagger}(s)$ requires other trajectories starting at state s. That is, we need multiple trajectories starting at the same state with different follower actions to estimate $\nabla_\theta A_F^{\theta\dagger}(s,b)$. This assumption is challenging to satisfy when the state-action space is either discrete but large or continuous. To tackle this challenge, Thoma et al. (2024) rely on an oracle that generates trajectories starting at an arbitrary state-action pair. However, the availability of such an oracle is also difficult to guarantee, especially when the leader does not have full control of the follower.

With our hypergradient formula (5), we do not need to estimate $\nabla_\theta V_F^{\theta\dagger}(s)$, and thus do not need multiple trajectories. Instead, we require estimating the Benefit $B_L^{\theta\dagger}(s,b)$, which we argue is easier to estimate, as it does not require multiple trajectories or an oracle. The equivalence between (7) and the third term in (5) is described by the Boltzmann Covariance trick, that is:

$$\begin{aligned}
&\mathbb{E}_{b\sim g^{\theta\dagger}(\cdot|s)}\left[Q_L^{\theta\dagger}(s,b)\nabla_\theta A_F^{\theta\dagger}(s,b)\right]\\
&=\mathbb{E}_b\left[Q_L^{\theta\dagger}(s,b)\cdot\left(\nabla_\theta Q_F^{\theta\dagger}(s,b)-\mathbb{E}_b\left[\nabla_\theta Q_F^{\theta\dagger}(s,b)\right]\right)\right]\\
&=\mathrm{Cov}_b\left[Q_L^{\theta\dagger}(s,b),\nabla_\theta Q_F^{\theta\dagger}(s,b)\;\middle|\; s\right]\\
&=\mathbb{E}_b\left[\left(Q_L^{\theta\dagger}(s,b)-\mathbb{E}_b\left[Q_L^{\theta\dagger}(s,b)\right]\right)\cdot\nabla_\theta Q_F^{\theta\dagger}(s,b)\right]\\
&=\mathbb{E}_{b\sim g^{\theta\dagger}(\cdot|s)}\left[B_L^{\theta\dagger}(s,b)\nabla_\theta Q_F^{\theta\dagger}(s,b)\right],
\end{aligned}$$

where $\mathrm{Cov}_b[\cdot,\cdot\mid s]$ is a covariance with respect to $b\sim g^{\theta\dagger}(\cdot\mid s)$. Because this transformation is an algebraic identity, replacing one form with the other preserves the expectation and therefore, preserves unbiasedness of the corresponding Monte Carlo estimator. We refer to this manipulation as the Boltzmann Covariance trick; the name is inspired by the Boltzmann Covariance Theorem (Movellan 1998), while the derivation here follows directly from the above identity under the nested optimization structure where the lower-level optimum is Boltzmann. Additionally, it leads to the interpretation of the third term in (5) as enhancing the probability of the follower's actions that are favorable to the leader.

Hypergradient for 2-Player Markov Game We extend Theorem 1 to 2-player MGs. Analogous to the policy gradient theorem for single-agent MDPs, we introduce the discounted state visitation distribution $d_\gamma^{\theta g}(s)$ (defined in Appendix B.1). For conciseness, we denote $d_\gamma^{\theta g^{\theta\dagger}}$ as $d_\gamma^{\theta\dagger}$. The policy gradient theorem for single-agent MDPs (Sutton et al. 1999; Sutton and Barto 2018) can be adapted to the leader's objective under a fixed follower as:

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

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

The following theorem extends the policy gradient theorem to 2-player MGs, providing the hypergradient for a stochastic leader policy. Note that, in general, the optimal leader policy in 2-player MGs may be stochastic. In this context, 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:

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

Theorem 2 (Hypergradient for MGs)**.** *The hypergradient of* (4) *with respect to* θ *is given by*

$$\nabla_\theta J_L(\theta,g^{\theta\dagger})=\frac{1}{1-\gamma_L}\mathbb{E}\Bigg[Q_L^{\theta\dagger}(s,a,b)\nabla_\theta\log f_\theta(a\mid s)+\frac{B_L^{\theta\dagger}(s,a,b)}{\beta}\nabla_\theta Q_F^{\theta\dagger}(s,a,b)\Bigg],\quad(8)$$

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*

$$\nabla_\theta Q_F^{\theta\dagger}(s,a,b)\qquad(9)$$
$$=\mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma_F^t V_F^{\theta\dagger}(s_t,a_t)\nabla_\theta\log f_\theta(a_t\mid s_t)\;\middle|\;\begin{matrix}s_0=s\\a_0=a\\b_0=b\end{matrix}\right].$$

The first term of (8) corresponds to the policy gradient theorem (Sutton et al. 1999; Sutton and Barto 2018), which reflects the improvement of the leader's utility by changing its own policy. The second term reflects the improvement in the leader's utility resulting from the change in the follower's best response due to the leader's policy update.

Implementation We propose Actor-Critic algorithms for both configurable MDP and MG settings, termed *Boltzmann Covariance HyperGradient (BC-HG)*. These algorithms comprise standard critic updates to estimate the current leader's Q-function using trajectories sampled under the best response, and actor updates utilizing a hypergradient estimated via the Benefit, based on (5) and (8). While our algorithm shares similarities with that of Thoma et al. (2024), it distinguishes itself by leveraging the Benefit. As discussed previously, this distinction is crucial for practical tractability. Algorithm 1 provides an overview of the proposed method.

We proceed under the following assumption, which allows us to focus on hypergradient estimation without the computational burden of estimating the follower's policy and value function:

Assumption 1 (White-box follower)**.** *The follower's policy and the value functions resulting from its learning process are accessible to the leader.*

Although our overall framework is designed for a decentralized learning setting, where the leader cannot directly intervene in the follower's updates, this constraint does not necessarily imply a black-box follower setting. Assumption 1 is often justifiable when the follower's best response is determined analytically with a task-specific method; accessing the best response is natural, but we do not want to change the follower's optimization process. This setup holds regardless of whether the follower is centralized or decentralized. This scenario is specifically addressed in our experiments (Sections 5.2 and 5.4). Furthermore, even if the follower is not strictly white-box, its policy and values can often be accurately estimated from interaction samples, mitigating the practical difference in many real-world applications.

We decompose the hypergradient into two components: the partial derivative $\partial_\theta J_L$ and the guiding term $\beta^{-1}\mathbb{E}[B_L^{\theta\dagger}\nabla_\theta Q_F^{\theta\dagger}]$. The partial derivative $\partial_\theta J_L$ corresponds to the first two terms of the hypergradient in configurable MDPs (5), and to the first term in MGs (8). The leader's value function $V_L^{\theta\dagger}$[2] can be computed via the Bellman equation using the estimated critic $Q_L^{\theta\dagger}$ and the best response $g^{\theta\dagger}$. The guiding term, corresponding to the final term of

---

[2] In Line 10 of Algorithm 1, we slightly abuse notation by defining $V_L(\bar{s})$ as the discounted cumulative expected leader reward conditioned on the initial state-action $\bar{s}=(s,a)$ in MG settings.

Algorithm 1: The BC-HG Algorithm

```
Initialize leader parameters $\theta^0$ and critic $Q_L^0$
for $i = 0$ to $N-1$ do
    (Follower computes best response $g^{\theta^i\dagger}$ for $\mathcal{M}_{\theta^i}$)
    Sample a batch of trajectories $\mathcal{D} \leftarrow \{\tau_j\}$ of length $T$
    Get follower's best response $g^{\theta^i\dagger}$ and value function $V_F^{\theta^i\dagger}$
    $Q_L^{i+1} \leftarrow$ CriticUpdate($Q_L^i, \mathcal{D}$) (Algorithm 2,3)
    # $\bar{s}$ denotes state $s$ for configurable MDPs
    # $\bar{s}$ denotes state-action pair $(s, a)$ for MGs
    for $(\bar{s}, b) \in \tau \in \mathcal{D}$ do
        $V_L(\bar{s}) \leftarrow \mathbb{E}_{b\sim g^{\theta^i\dagger}(\cdot|\bar{s})}\left[Q_L^{i+1}(\bar{s}, b)\right]$
        $B_L(\bar{s}, b) \leftarrow Q_L^{i+1}(\bar{s}, b) - V_L(\bar{s})$
        $\mathcal{D}_{\bar{s}b} \leftarrow \left\{\tau^{k:T} \mid (\bar{s}_k, b_k) = (\bar{s}, b) \text{ for } (\bar{s}_k, b_k) \in \tau \in \mathcal{D}\right\}$
        $\widehat{\nabla_\theta Q_F}(\bar{s}, b) \leftarrow$ FollowerQGrad($\mathcal{D}_{\bar{s}b}, \theta^i, V_F^i$) (Algorithm 6,7)
    end for
    $\widehat{\partial_\theta J_L} \leftarrow$ PartialDerivative($\mathcal{D}, \theta^i, V_L, Q_L^{i+1}$) (Algorithm 4,5)
    $\widehat{\nabla_\theta J_L} \leftarrow \widehat{\partial_\theta J_L} + \frac{1}{\beta|\mathcal{D}|}\sum_{\tau\in\mathcal{D}}\sum_{(\bar{s}_t, b_t)\in\tau} \gamma_L^t B_L(\bar{s}_t, b_t)\widehat{\nabla_\theta Q_F}(\bar{s}_t, b_t)$
    $\theta^{i+1} \leftarrow \theta^i + \alpha\widehat{\nabla_\theta J_L}$
end for
```

each hypergradient, is estimated by averaging the gradient of the follower's Q-function, $\nabla_\theta Q_F^{\theta\dagger}(s, b)$ (or $\nabla_\theta Q_F^{\theta\dagger}(s, a, b)$), weighted by the Benefit $B_L^{\theta\dagger}(s, b)$ (or $B_L^{\theta\dagger}(s, a, b)$), over the sampled transitions. The gradient $\nabla_\theta Q_F^{\theta\dagger}$, as shown in (6) (or (9)), is estimated as the discounted cumulative gradient along a trajectory segment following each transition (s, b) (or (s, a, b)) sampled for the outer expectation in (5) (or (8)).

5 Experiments

We demonstrate the advantages of our proposed BC-HG over baseline methods designed for configurable MDPs and 2-player MGs in both discrete and continuous state settings.

5.1 Configurable MDPs (Discrete)

Four-Rooms Task This task is adapted from Thoma et al. (2024). The follower can move up, down, right, or left (4 actions) within the level (104 states) visualized in Figure 1. The follower starts at the cell denoted S and receives a positive reward upon reaching the goal cell G. In contrast, the leader receives an immediate reward of $+1$ when the follower visits the target cell, which is denoted by "1" and highlighted in green. To guide the follower, the leader can place a negative incentive (penalty) in each cell, incurring a cost proportional to the total negative incentive placed when the follower reaches the goal. See Appendix C.2 for details.

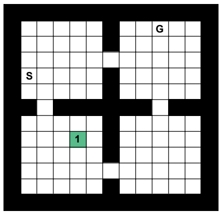


Figure 1: Four-Rooms Environment

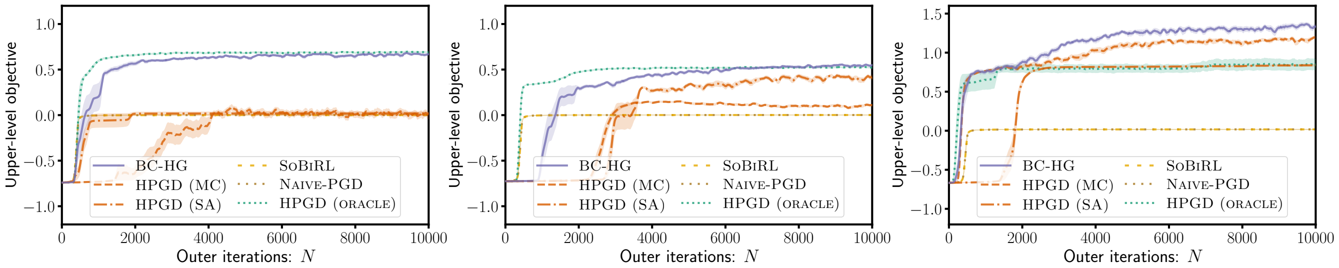
Baselines We compare the proposed approach against the following baselines: Naive-PGD, which employs first-order policy gradient updates (Rajeswaran, Mordatch, and Kumar 2020; Vu et al. 2022) ignoring changes in the follower's best response (i.e., using $\partial_\theta J_L(\theta, g) \mid_{g=g^{\theta\dagger}}$); SoBiRL, a hypergradient approach by Yang, Gao, and Yuan (2025), originally designed for reward shaping from human feedback; and HPGD, another hypergradient approach by Thoma et al. (2024). We evaluate three variations of HPGD: HPGD (ORACLE), which relies on an oracle for hypergradient estimation (as presented in the original paper); HPGD (MC), which uses Monte-Carlo estimation with a batch of trajectories collected during interaction (as implemented by the authors); and HPGD (SA), which utilizes SARSA-type estimation similar to our proposed approach. Note that HPGD (ORACLE) requires oracle access, a condition not assumed by the other methods. In this experiment, additional trajectories of size 10^4 per outer iteration are generated by the oracle. See Appendix C.1 for details.

Settings For all approaches, the follower's best response is computed via soft value iteration with entropy regularization coefficients $\beta \in \{1\times10^{-3}, 3\times10^{-3}, 5\times10^{-3}\}$. For the main results in Figure 2, we use a batch size of 100, which is small relative to the state-space size of 104. For each approach, we perform a grid search over hyperparameters and select the best combination based on average performance across 10 random seeds. Results with different batch sizes (200, 400, 1000) are reported in Appendix C.2.

Results The results are presented in Figures 2c, 2b, and 2a for varying entropy regularization parameters β. The proposed approach, BC-HG, consistently outperforms the baselines. Generally, the performance of HPGD variants (excluding the oracle version) degrades as entropy regularization weakens (i.e., smaller β). With a smaller β, selecting different actions at the same state becomes less likely. Consequently, the core assumption of HPGD variants, that trajectories starting at the same state with different actions are collected during interaction, becomes harder to satisfy. Results with different batch sizes are shown in Appendix C.2. The superiority of BC-HG over HPGD (SA) underscores the impact of the Boltzmann Covariance trick, as the only difference lies in the guiding term's form in the hypergradients. SoBiRL, Naive-PGD, and HPGD variants with $\beta = 1 \times 10^{-3}$ become stuck in a local minimum where no negative incentive is placed, resulting in zero penalty and zero positive reward for the leader. This occurs because the third term of the hypergradient in Theorem 1, capturing the shift in the best response, is poorly estimated, emphasizing the penalty effect in the first term.

5.2 Configurable MDPs (Continuous)

Task Description and Settings Inspired by Afram and Janabi-Sharifi (2014), we designed an *n-zone building ther-*



(a) $\beta = 1 \times 10^{-3}$, BatchSize $= 100$ (b) $\beta = 3 \times 10^{-3}$, BatchSize $= 100$ (c) $\beta = 5 \times 10^{-3}$, BatchSize $= 100$

Figure 2: Results on Four-Rooms Task (BatchSize $= 100$).

mal control task where the follower controls heating, ventilation, and air conditioning (HVAC) units to minimize temperature deviations in n zones using a linear quadratic regulator (LQR). The leader aims to enhance temperature stability and energy efficiency by adjusting building parameters, such as insulation and airflow.

This task is formulated as a bi-level extension of infinite-horizon discounted LQR with entropy regularization. The state $s \in \mathbb{R}^n$ is temperature deviations, with transitions defined as: $s_{t+1} = A_\theta s_t + Bb_t + w_t$, where $w_t \sim \mathcal{N}(0, W)$, $A_\theta \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $W \in \mathbb{R}^{n\times n}$. m is the number of HVAC units. The follower's reward function is: $r_F(s, b) := -s^\top \bar{Q} s - b^\top \bar{R} b$, where $\bar{Q} \in \mathbb{R}^{n\times n}$ is positive semi-definite and $\bar{R} \in \mathbb{R}^{m\times m}$ is positive definite. The leader's parameter $\theta \in \mathbb{R}^{2n}$ represents the insulation level of n zones and the airflow level of n inter-zone ventilations.

The number of zones and HVAC units is set to $n = 4$ and $m = 2$, respectively. The follower's best response is computed via Riccati iteration for all approaches (see Appendix C.4 for details). We conduct a grid search over hyperparameters for each approach using 10 random seeds. Further details are provided in Appendix C.5.

Baselines We consider the following baselines: Naive-PGD, HPGD (MC), and HPGD (TD). In continuous state settings, Thoma et al. (2024) implement HPGD by estimating the follower's value function gradient using a function approximator rather than additional trajectories, estimating the gradient as a vector-valued function in the cumulative form of (6). This implementation does not rely on an oracle. However, the learning cost of the vector-valued function increases with higher-dimensional leader parameters. HPGD estimates the leader's Q-function via Monte-Carlo estimation in HPGD (MC) and via the TD-method with neural networks in HPGD (TD). Note that SoBiRL is inapplicable here because the leader parameterizes the transitions.

Results Figure 3 displays the results. Note that HPGD (oracle) uses additional trajectories of size 10^4 per outer iteration generated by the oracle. BC-HG achieves the highest convergence in the fewest iterations. Naive-PGD shows more gradual improvement, suggesting that while partial derivative information is useful in this non-competitive task, it is insufficient without hypergradient information. HPGD exhibits slower improvement or even degradation with certain hyperparameters. These results imply that the estimation accuracy of the follower's value gradient was insufficient for the 8-dimensional leader parameter. Given that the learning cost of the value gradient increases with the dimensionality of the leader parameter, BC-HG is more scalable. See Appendix C.5 for more results.

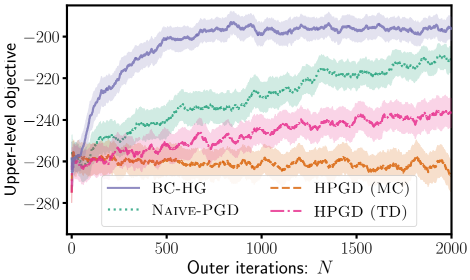
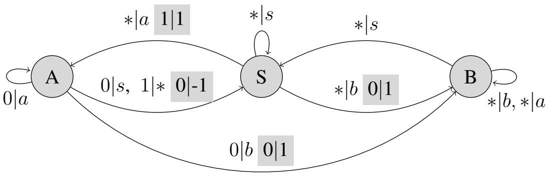


Figure 3: Results on Building Thermal Control.

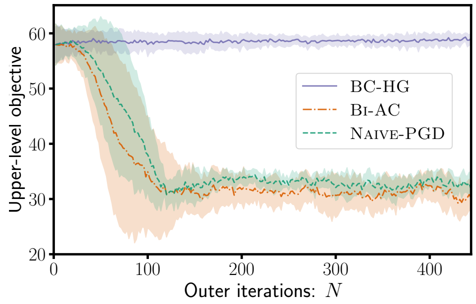
5.3 2-Player Markov Games (Discrete)

Task Description and Setting This task involves finite state and action spaces with deterministic transitions. The state space is $\mathcal{S} = \{S, A, B\}$, and action spaces are $\mathcal{A} = \{0, 1\}$ (leader) and $\mathcal{B} = \{s, a, b\}$ (follower). Deterministic transition dynamics and reward functions are illustrated in Figure 4a. Leader actions at states S and B do not influence rewards or transitions; thus, we focus on the leader's action selection probability at state A. If $f_\theta(0 \mid A) \approx 1$, the follower prefers actions a, b, and s at states S, A, and B, respectively. This results in the cycle $S \xrightarrow{*|a} A \xrightarrow{0|b} B \xrightarrow{*|s} S$, yielding rewards of $1, 2, 0$ for the follower and $1, 0, 0$ for the leader. Conversely, if $f_\theta(1 \mid A) \approx 1$, the follower prefers actions b, b, and s at states S, A, and B, respectively. This leads to the cycle $S \xrightarrow{*|b} B \xrightarrow{*|s} S$, yielding rewards of $1, 0$ for the follower and $0, 0$ for the leader. Since the former scenario yields a higher return for the follower, the follower prefers action a at S even when $f_\theta(0 \mid A) = p \in (0, 1)$, despite the risk of a negative reward -1 with probability $1 - p$. From the leader's perspective, the $S \leftrightarrow A$ cycle yields the highest cumulative reward. Therefore, the optimal leader strategy is to minimize p while maintaining a high probability of the follower selecting action a at S.

For all approaches, the follower's best response is computed via soft Q-iteration with entropy regularization coeffi-



(a) State Transition Diagram. Nodes represent states and edges represent transitions. Labels on edges indicate $\mathcal{A} \mid \mathcal{B}$ and $r_L \mid r_F$. The absence of shaded labels implies zero rewards. "$*$" represents a wildcard.



(b) Result

Figure 4: Task Description and Results on ToyMarkovGame.

cient $\beta = 5 \times 10^{-2}$. We conduct a grid search over hyperparameters (number of critic/actor updates per outer iteration, on-policy vs. off-policy sampling) and select the best combination based on average performance across 10 random seeds. All learning rates are fixed. Performance is evaluated by averaging the cumulative leader reward across 10 rollouts. See Appendix C.3 for further details.

Baselines We consider two baseline approaches. The first is the first-order policy gradient, which ignores the second term of the hypergradient in Theorem 2. The second approach is based on Bi-AC (Zhang et al. 2020), originally proposed for centralized training, which we adapt to our decentralized setting. The key idea of Bi-AC is to update the leader's action-value function Q_L as:

$$\begin{aligned} Q'_L(s,a,b) \leftarrow\ & (1-\alpha) Q_L(s,a,b) \\ & + \alpha(r_L(s,a,b) + \gamma_L Q_L(s', a^*, b^*(a^*))), \end{aligned}$$

where $b^*(a') = \operatorname{argmax}_{b'} Q_F^{\theta\dagger}(s', a', b')$ is the best response at state s' given leader action a', and $a^* = \operatorname{argmax}_{a'} Q_L(s', a', b^*(a')))$ is the best leader action under the follower's best response. Essentially, at each state, the Stackelberg equilibrium of the normal-form 2-player game with utility matrices $(Q_L(s', \cdot, \cdot), Q_F^{\theta\dagger}(s', \cdot, \cdot))$ is derived. We replace the action-value function update in the first baseline with this Stackelberg game-based update. Further details are provided in Appendix C.1.

Results Figure 4b shows the results. BC-HG clearly outperforms the baselines. BC-HG maintains $f_\theta(0 \mid A) \approx 0.53$, with the corresponding follower best response $g^{\theta\dagger}(a \mid S, *) \approx 1$. Consequently, the $S \to A$ loop and $S \to A \to B$ loop occur with probabilities around 0.47 and 0.53, respectively. In contrast, baseline approaches maintain $f_\theta(0 \mid A) \approx 0.4$ and $g^{\theta\dagger}(\cdot \mid S, *) \approx 0, 0.47, 0.53$ for s, a, b, respectively. As $f_\theta(0 \mid A)$ is too small, the $S \leftrightarrow B$ loop occurs, resulting in a loss for the leader. This trend persists across a wide range of hyperparameters, as discussed in Appendix C.3. The policy gradient with Bi-AC-type Q-update exhibits a similar trend.

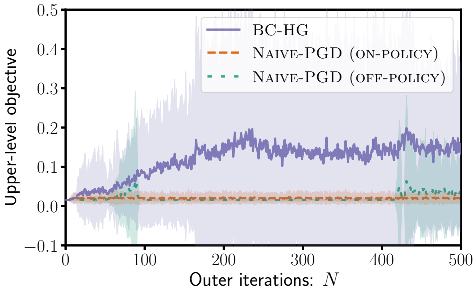


Figure 5: Results on the Bi-Level LQR Task.

5.4 2-Player Markov Games (Continuous)

Bi-Level LQR in 2-Player Markov Games We consider the extension of LQR tasks to MGs. Introducing the leader's action, the state transition is defined as: $s_{t+1} = As_t + Bb_t + Ca_t$, where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, and $C \in \mathbb{R}^{n \times k}$ are constant. The follower's reward function is $r_F(s,a,b) = -s^\top \bar{Q} s - b^\top \bar{R} b$. The leader's policy $f_\theta(a \mid s)$ is modeled by a Gaussian distribution $\mathcal{N}(K_\theta s, W)$, where $K_\theta \in \mathbb{R}^{k \times n}$ is the leader's policy parameter and $W \in \mathbb{R}^{k \times k}$ is the fixed covariance matrix. In this experiment, the leader's reward function is:

$$r_L(s,a,b) := \exp\Big(-\frac{1}{2}(s - s^\star)^\top \Sigma (s - s^\star) \Big),$$

where $\Sigma \in \mathbb{R}^{n \times n}$ defines the reward sharpness. The leader aims to guide the follower to visit $s^\star \in \mathcal{S}$. The follower's best response is derived by solving the Riccati equation. Further details are provided in Appendix C.6.

Results Figure 5 presents the results for the bi-level LQR task. BC-HG outperforms the baseline. A possible reason for the high variance in the upper-level performance of the proposed approach is that leader policies achieving higher cumulative rewards tend to exhibit higher variance in cumulative reward, as demonstrated in Figure 10 (Appendix C.6).

6 Conclusion

We developed hypergradient-based bi-level RL algorithms for configurable MDPs and MGs. By leveraging the Boltzmann covariance trick, we circumvented the need for oracle-based trajectory generation while simultaneously enabling efficient and unbiased hypergradient estimation. In numerical experiments, our approach demonstrated scalability to large state spaces and high-dimensional parameterizations in configurable MDPs. We also validated the effectiveness of our method in Markov game settings with both discrete and continuous state spaces. Future work includes addressing more realistic decentralized follower scenarios where the follower is either suboptimal or black-box.

References

Afram, A.; and Janabi-Sharifi, F. 2014. Theory and Applications of HVAC Control Systems – A Review of Model Predictive Control (MPC). *Building and Environment*, 72: 343–355.

Chakraborty, S.; Bedi, A. S.; Koppel, A.; Manocha, D.; Wang, H.; Huang, F.; and Wang, M. 2023. PARL: A Unified Framework for Policy Alignment in Reinforcement Learning from Human Feedback. In *International Conference on Learning Representations*.

Hu, J.; and Wellman, M. P. 2003. Nash Q-Learning for General-Sum Stochastic Games. *J. Mach. Learn. Res.*, 4: 1039–1069.

Metelli, A. M.; Mutti, M.; and Restelli, M. 2018. Configurable Markov Decision Processes. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 3491–3500. PMLR.

Movellan, J. R. 1998. A Learning Theorem for Networks at Detailed Stochastic Equilibrium. *Neural Computation*, 10(5): 1157–1178.

Muneeb, S. M.; Adhami, A. Y.; Asim, Z.; and Jalil, S. A. 2019. Bi-Level Decision Making Models for Advertising Allocation Problem under Fuzzy Environment. *International Journal of System Assurance Engineering and Management*, 10(2): 160–172.

Rajeswaran, A.; Mordatch, I.; and Kumar, V. 2020. A Game Theoretic Framework for Model Based Reinforcement Learning. In Iii, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, 7953–7963. PMLR.

Shen, H.; Yang, Z.; and Chen, T. 2024. Principled Penalty-based Methods for Bilevel Reinforcement Learning and RLHF. In Salakhutdinov, R.; Kolter, Z.; Heller, K.; Weller, A.; Oliver, N.; Scarlett, J.; and Berkenkamp, F., eds., *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, 44774–44799. PMLR.

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

Sutton, R. S.; McAllester, D.; Singh, S.; and Mansour, Y. 1999. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Proceedings of the 13th International Conference on Neural Information Processing Systems (NeurIPS)*, NIPS'99, 1057–1063. MIT Press.

Thoma, V.; Pásztor, B.; Krause, A.; Ramponi, G.; and Hu, Y. 2024. Contextual Bilevel Reinforcement Learning for Incentive Alignment. In *The 38th Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Vu, Q.-L.; Alumbaugh, Z.; Ching, R.; Ding, Q.; Mahajan, A.; Chasnov, B.; Burden, S.; and Ratliff, L. J. 2022. Stackelberg Policy Gradient: Evaluating the Performance of Leaders and Followers. In *ICLR 2022 Workshop on Gamification and Multiagent Solutions*.

Yang, B.; Zheng, L.; Ratliff, L. J.; Boots, B.; and Smith, J. R. 2023. Stackelberg Games for Learning Emergent Behaviors During Competitive Autocurricula. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 5501–5507. IEEE.

Yang, Y.; Gao, B.; and Yuan, Y.-X. 2025. Bilevel Reinforcement Learning via the Development of Hyper-gradient without Lower-Level Convexity. In *International Conference on Artificial Intelligence and Statistics*, 4780–4788. PMLR.

Zhang, H.; Chen, W.; Huang, Z.; Li, M.; Yang, Y.; Zhang, W.; and Wang, J. 2020. Bi-Level Actor-Critic for Multi-Agent Coordination. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34: 7325–7332.

Zheng, Z.; and Gu, S. 2025. Safe Multiagent Reinforcement Learning With Bilevel Optimization in Autonomous Driving. *IEEE Transactions on Artificial Intelligence*, 6(4): 829–842.