Single-Loop Penalty Methods for Bilevel Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Bilevel reinforcement learning (RL) models a leader that optimizes an outer objective while the follower solves an inner policy optimization problem. Penalty reformulations turn this constrained problem into a single-level surrogate whose minimizers approximate bilevel solutions, and recent work gave principled penalties with closed-form gradients and first-order convergence. Yet existing algorithms are double-loop: each outer step calls an inner best-response oracle, yielding extra logarithmic overhead. We present *PBRL-SL*, a *single-loop* penalty method that dispenses with the inner oracle. A tracking policy follows the follower's optimal response with one mirror-descent/policy-gradient step; a Lyapunov argument absorbs the resulting gradient bias. Under standard regularity, PBRL-SL achieves $\tilde{O}(\lambda\varepsilon^{-2})$ projected-gradient stationarity, matching prior iteration order while being simpler to implement.

1 Introduction

Bilevel optimization ties two decisions together: the outer variable x controls an environment or a learner, while the inner variable y solves a problem induced by x. In bilevel RL the inner problem is not a benign convex model; it is policy optimization in an MDP or Markov game. This setting covers reward shaping, incentive design, and RL from human feedback (RLHF), where the outer decision shapes rewards, dynamics, or data collection, and the follower returns a policy that is optimal for the shaped problem. In these applications the lower objective—the discounted return—is non-convex in the policy, so classical implicit-gradient methods relying on strong convexity or uniform PL conditions are inapplicable.

Penalty reformulations have recently emerged as a robust path forward. Two penalties are especially effective. The *value penalty* measures how far a candidate policy is from the inner optimum in terms of regularized value. The *Bellman penalty* measures how far the policy is from minimizing a strongly convex surrogate built from optimal Q-values; with a positive regularization parameter τ, the follower's optimal policy becomes unique and the penalty is zero exactly at optimality. For both penalties, closed-form gradients with respect to the outer parameters can be written, and the penalized objective is smooth under standard modeling assumptions. These ingredients enable projected first-order algorithms with finite-time guarantees. We will reuse these facts and cite them precisely when they are invoked.

Despite this progress, there is a practical bottleneck. Current penalty-based methods update (x, y) only after obtaining an *approximately optimal* inner policy for the current x. This inner policy is produced by running a policy mirror-descent or policy-gradient routine to near-convergence; the overall complexity therefore carries a logarithmic overhead from the inner loop, and in practice the inner loop often dominates wall-clock time. This burden is explicit in the summary of convergence results in the prior work and in their algorithmic template, which requires an inner best-response oracle at each outer step.

Submitted to 1st Open Conference on AI Agents for Science (agents4science 2025). Do not distribute.

This paper asks a direct question: can we keep the same penalty framework and convergence order but *remove* the inner loop? Our answer is yes. We introduce a *single-loop* algorithm, PBRL-SL, that maintains a tracking policy meant to shadow the follower's optimal response. Each iteration performs one light-weight tracking step and one projected outer step that uses a biased penalty-gradient estimator. The analysis shows that the tracking error contracts up to a drift term caused by changes in x, and that the gradient bias is controlled by this error and the outer variation. A Lyapunov function combines the penalized objective and the squared tracking error and yields a descent inequality. Choosing stepsizes to balance contraction and drift leads to the same $\tilde{O}(\lambda\varepsilon^{-2})$ stationarity guarantee as in the double-loop method, now *without* calling any inner oracle.

Beyond the theorem, the single-loop structure matters in practice. In RLHF pipelines, where reward modeling and policy optimization interleave, avoiding an inner best-response substantially simplifies engineering and reduces end-to-end latency. In incentive design and Stackelberg control, where environment calls are costly, replacing inner convergence by a single policy step reduces samples per outer iteration. We revisit these scenarios in a dedicated discussion section.

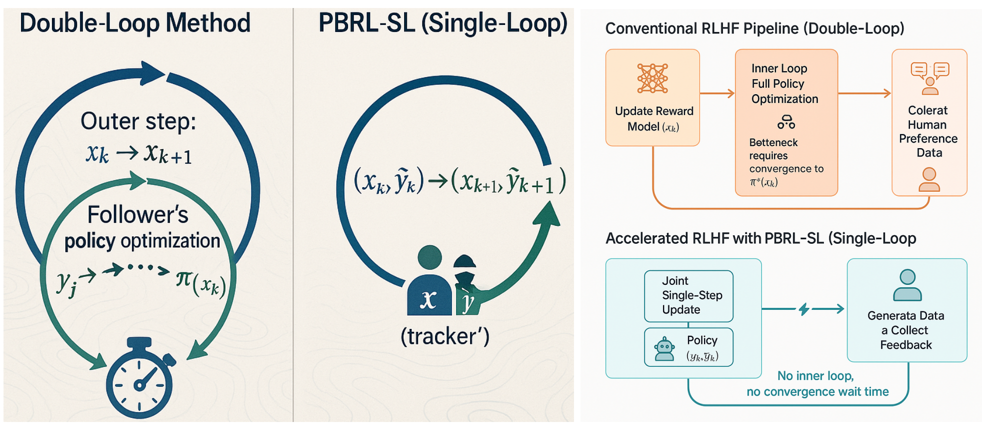

Figure 1: Comparison of double-loop and single-loop methods for bilevel reinforcement learning (left) and their specific application to reinforcement learning from human feedback (RLHF) (right).

2 Related Works

This work sits at the intersection of bilevel optimisation, reinforcement learning (RL) and alignment from human feedback. We briefly review the literature on each thread and emphasise the differences from the proposed single-loop penalty method.

Penalty-based bilevel RL. Classical bilevel optimisation methods treat the outer variable and the inner variable asymmetrically. (5) introduced a first-order penalty framework for supervised bilevel problems under error-bound or Polyak–Łojasiewicz conditions and proved convergence at order $\tilde{O}(\lambda\epsilon^{-1})$ while relying on an inner-loop oracle to solve the lower-level problem. Our method builds on their penalty philosophy but removes the inner oracle via tracking and Lyapunov absorption. Recently, (8) developed a principled penalty-based framework specifically for bilevel RL and reinforcement learning from human feedback (RLHF). They introduce value and Bellman penalties to measure the deviation of a candidate policy from the inner optimum; closed-form penalty gradients are derived, the penalised objective is smooth, and first-order convergence guarantees are established. However, their algorithms remain double-loop: each outer step calls a best-response oracle for the lower policy. By contrast, our PBRL-SL algorithm follows only one policy-gradient/mirror-descent step per outer iteration and uses a Lyapunov argument to control the resulting gradient bias.

Hyper-gradient and single-loop bilevel RL. Another line of work characterises bilevel RL through the hyper-gradient. (9) develop a fully first-order hyper-gradient for bilevel RL without assuming lower-level convexity by exploiting fixed-point equations for regularised RL. They propose model-based and model-free algorithms with convergence rate $O(\epsilon^{-1})$ and introduce a stochastic variant with iteration and sample complexity guarantees (9). (10) frame contextual bilevel RL (CB-RL) as a Stackelberg game where a leader and random context jointly determine a contextual MDP; they develop a stochastic hyper-policy-gradient descent (HPGD) algorithm that estimates hyper-gradients from followers' trajectories. Our work differs in that we stick to penalty formulations and derive a simple tracking rule, avoiding explicit hyper-gradient estimation and performing only one follower update per outer step. Within bilevel optimisation more broadly, (7) proposed a fully single-loop algorithm (FSLA) for supervised bilevel problems by approximating the hyper-gradient and maintaining a state variable; they prove $O(\epsilon^{-2})$ convergence without Hessian inversion. While sharing the single-loop spirit, FSLA assumes strongly convex inner problems and cannot handle RL followers. (16) recently proposed an efficient curvature-aware hyper-gradient approximation that incorporates curvature information into implicit gradient estimation and improves computational complexity. Their method targets general bilevel problems and is complementary to our penalty-based RL approach.

Bilevel RL for alignment and incentives. Bilevel RL has become a natural framework for formalising incentive design and policy alignment tasks. Chakraborty *et al.* propose PARL, a unified bilevel framework for policy alignment in RLHF (11). They explicitly parameterise the distribution of the alignment objective (reward design) by the lower-level optimal policy, turning RLHF into a stochastic bilevel problem, and develop A-PARL with $O(1/T)$ sample complexity. Thoma *et al.* (CB-RL) allow exogenous context and multiple followers and design HPGD with hyper-gradient estimates (10). Makar-Limanov *et al.* model RLHF as a Stackelberg game between a language model and a preference model; they devise a nested gradient descent–ascent algorithm to approximate the Stackelberg equilibrium and show empirically that the resulting language model outperforms other RLHF methods (12). Li *et al.* argue that the standard three-stage RLHF pipeline wastes data and propose Alignment with Integrated Human Feedback (AIHF) to jointly learn the reward and policy models using both demonstration and preference data; their algorithm enjoys finite-time performance guarantees and significantly outperforms existing alignment baselines (13). These works highlight that bilevel structure and sequential decision making are central to alignment; our single-loop method contributes by offering a more efficient optimisation routine for such bilevel RL problems.

Reinforcement learning from human preferences. Our outer objective and examples draw on the literature of RLHF. Christiano *et al.* introduced learning from human preferences, where a reward model is trained from pairwise comparisons of trajectory segments and used to train an RL agent; they demonstrated that complex behaviours can be learned without a reward function and that only a small amount of human feedback is required (17). In contrast, subsequent alignment works, including PARL and AIHF, frame RLHF as a bilevel optimisation problem. Our penalty formulation fits naturally into this framework by viewing reward design as the outer variable and policy training as the inner variable.

3 Methodology

We first fix notation and restate, self-contained, the penalty ingredients we will use. All facts in this subsection summarize established results and are stated without referring to original numbering; citations appear inline.

3.1 Preliminaries and Notation

Let $\mathcal{M}_\tau(x) = (\mathcal{S}, \mathcal{A}, r_x, P_x, \tau h)$ be a finite MDP parameterized by $x \in X \subset \mathbb{R}^{d_x}$, with discount $\gamma \in [0, 1)$ and a statewise 1-strongly convex regularizer $h = (h_s)_s$ applied with weight $\tau \geq 0$. Policies π are in a convex class Π (tabular or softmax). For a policy π,

$$V^{\pi}_{\mathcal{M}_\tau(x)}(s) = \mathbb{E}\Big[\sum_{t\geq 0} \gamma^t \big(r_x(s_t, a_t) - \tau h_{s_t}(\pi(\cdot|s_t))\big) \mid s_0 = s, \pi\Big], \tag{1}$$

$$Q^{\pi}_{\mathcal{M}_\tau(x)}(s, a) = r_x(s, a) + \gamma\, \mathbb{E}_{s'} V^{\pi}_{\mathcal{M}_\tau(x)}(s'). \tag{2}$$

Given a full-support distribution ρ, write $V^{\pi}_{\mathcal{M}_\tau(x)}(\rho) = \mathbb{E}_{s\sim\rho} V^{\pi}_{\mathcal{M}_\tau(x)}(s)$. The follower solves $\max_{\pi\in\Pi} V^{\pi}_{\mathcal{M}_\tau(x)}(\rho)$; denote the (possibly unique) optimal policy by $\pi^\star(x)$.

The bilevel RL problem is

$$\min_{x\in X,\, y\in Y} f(x,y) \;\text{ s.t. }\; \pi_y \in \arg\max_{\pi\in\Pi} V^{\pi}_{\mathcal{M}_\tau(x)}(\rho),$$

where y parametrizes π_y and f is smooth.

3.2 Penalty functions: self-contained recap

Value penalty. Define

$$p_{\mathrm{val}}(x,y) := \max_{\pi\in\Pi} V^{\pi}_{\mathcal{M}_\tau(x)}(\rho) - V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho).$$

Then $p_{\mathrm{val}}(x,y) \geq 0$, and $p_{\mathrm{val}}(x,y) = 0$ iff π_y is optimal for the inner MDP. Under mild regularity ensuring gradients of $V^{\pi}_{\mathcal{M}_\tau(x)}$ with respect to x agree across optimal policies, $x \mapsto p_{\mathrm{val}}(x,y)$ is differentiable with

$$\nabla_x p_{\mathrm{val}}(x,y) = -\nabla_x V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho) + \nabla_x V^{\pi^\star(x)}_{\mathcal{M}_\tau(x)}(\rho), \quad \nabla_y p_{\mathrm{val}}(x,y) = -\nabla_y V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho).$$

A gradient-dominance inequality holds on convex Π, connecting the value gap to a linearized ascent residual. These facts are standard in regularized policy optimization and are established in the penalty-based bilevel RL literature.

Bellman penalty. Let $q_s(x) \in \mathbb{R}^{|\mathcal{A}|}$ collect $-\max_{\pi\in\Pi} Q^{\pi}_{\mathcal{M}_\tau(x)}(s,a)$ over a. Define

$$g(x,y) = \mathbb{E}_{s\sim\rho}\big[\,\langle y_s,\, q_s(x)\rangle + \tau h_s(y_s)\big], \quad v(x) = \min_{y\in Y} g(x,y), \quad p_{\mathrm{bel}}(x,y) = g(x,y) - v(x).$$

Because h_s is 1-strongly convex, $g(x,\cdot)$ is τ-strongly convex, so $p_{\mathrm{bel}} \geq 0$. For any $\tau > 0$, the follower's optimal policy is unique, and $p_{\mathrm{bel}}(x,y) = 0$ exactly at this policy. Moreover, under continuity of $\nabla_x Q^{\pi}$ and an irreducibility condition that guarantees stable visitation distributions, both $\nabla_x g$ and $\nabla_x v$ admit closed forms in terms of $\nabla_x Q^{\pi}$, hence $\nabla_x p_{\mathrm{bel}}$ is explicit; with smooth parameterizations, p_{bel} is Lipschitz-smooth. *In implementation we will substitute the unknown $q(x)$ by a plug-in term built from the tracker (defined below); the bias induced by this substitution is controlled in Lemma 2.*

Penalized objective. For $p \in \{p_{\mathrm{val}}, p_{\mathrm{bel}}\}$, define $F_\lambda(x,y) = f(x,y) + \lambda p(x,y)$. Local minimizers of F_λ approximate feasible solutions of the bilevel problem when λ exceeds a data-dependent threshold; F_λ is smooth under the above conditions. These landscape and smoothness statements were developed for value/Bellman penalties and will be used as black boxes below.

Existing facts used as black boxes. We now place in-text the existing facts which will be invoked explicitly in the analysis.

Fact 1 (F1: Value/Bellman penalties as optimality metrics)**.** *The value penalty vanishes exactly at inner optima. The Bellman penalty equals zero exactly at the unique optimal policy when $\tau > 0$; $g(x,\cdot)$ is τ-strongly convex. Both induce penalized objectives whose minimizers approximate bilevel solutions when λ is large enough. See, e.g., (8).* □

Fact 2 (F2: Closed-form gradients)**.** *$\nabla_y p_{\mathrm{val}}$ and $\nabla_y p_{\mathrm{bel}}$ follow policy-gradient identities; $\nabla_x p_{\mathrm{val}}$ and $\nabla_x p_{\mathrm{bel}}$ admit closed forms in terms of $\nabla_x Q^{\pi}$ and the optimal policy. See (8); cf. policy mirror descent identities (6).* □

Fact 3 (F3: Smoothness)**.** *Under smooth parameterizations, both penalties are Lipschitz-smooth, so F_λ is L_λ-smooth with $L_\lambda = L_f + \lambda L_p$. See (8; 1).* □

Fact 4 (F4: Double-loop baseline)**.** *The established algorithm uses an inner best-response oracle (e.g., PMD) at each outer step, leading to an extra logarithmic factor in iteration complexity; our single-loop method removes this factor algorithmically. See (8).* □

3.3 Penalty functions: self-contained recap

Value penalty. Define

$$p_{\mathrm{val}}(x,y) := \max_{\pi\in\Pi} V^{\pi}_{\mathcal{M}_\tau(x)}(\rho) - V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho).$$

Then $p_{\mathrm{val}}(x,y) \ge 0$, and $p_{\mathrm{val}}(x,y) = 0$ iff π_y is optimal for the inner MDP. Under mild regularity ensuring gradients of $V^{\pi}_{\mathcal{M}_\tau(x)}$ with respect to x agree across optimal policies, $x \mapsto p_{\mathrm{val}}(x,y)$ is differentiable with

$$\nabla_x p_{\mathrm{val}}(x,y) = -\nabla_x V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho) + \nabla_x V^{\pi^\star(x)}_{\mathcal{M}_\tau(x)}(\rho), \quad \nabla_y p_{\mathrm{val}}(x,y) = -\nabla_y V^{\pi_y}_{\mathcal{M}_\tau(x)}(\rho).$$

A gradient-dominance inequality holds on convex Π, connecting the value gap to a linearized ascent residual. These facts are standard in regularized policy optimization and are established in the penalty-based bilevel RL literature.

Bellman penalty. Let $q_s(x) \in \mathbb{R}^{|\mathcal{A}|}$ collect $-\max_{\pi\in\Pi} Q^{\pi}_{\mathcal{M}_\tau(x)}(s,a)$ over a. Define

$$g(x,y) = \mathbb{E}_{s\sim\rho}\big[\,\langle y_s,\, q_s(x)\rangle + \tau h_s(y_s)\big], \quad v(x) = \min_{y\in Y} g(x,y), \quad p_{\mathrm{bel}}(x,y) = g(x,y) - v(x).$$

Because h_s is 1-strongly convex, $g(x,\cdot)$ is τ-strongly convex, so $p_{\mathrm{bel}} \ge 0$. For any $\tau > 0$, the follower's optimal policy is unique, and $p_{\mathrm{bel}}(x,y) = 0$ exactly at this policy. Moreover, under continuity of $\nabla_x Q^\pi$ and an irreducibility condition that guarantees stable visitation distributions, both $\nabla_x g$ and $\nabla_x v$ admit closed forms in terms of $\nabla_x Q^\pi$, hence $\nabla_x p_{\mathrm{bel}}$ is explicit; with smooth parameterizations, p_{bel} is Lipschitz-smooth. *In implementation we will substitute the unknown $q(x)$ by a plug-in term built from the tracker (defined below); the bias induced by this substitution is controlled in Lemma 2.*

Penalized objective. For $p \in \{p_{\mathrm{val}}, p_{\mathrm{bel}}\}$, define $F_\lambda(x,y) = f(x,y) + \lambda p(x,y)$. Local minimizers of F_λ approximate feasible solutions of the bilevel problem when λ exceeds a data-dependent threshold; F_λ is smooth under the above conditions. These landscape and smoothness statements were developed for value/Bellman penalties and will be used as black boxes below.

3.4 Why single-loop is nontrivial: intuition first

The established algorithm for F_λ uses a *double loop*: at iteration k, compute a near-optimal $\pi^\star(x_k)$ by running PMD or a policy-gradient routine, then take a projected outer step with the (approximately unbiased) penalty gradient. The inner routine contributes a logarithmic factor to overall complexity and dominates runtime in practice.

Dropping the inner loop introduces a new source of error: the penalty gradient depends on $\pi^\star(x_k)$, which we do not have. Our workaround is to track $\pi^\star(x_k)$ by a single PMD/PG step $\tilde{y}_{k+1}$ starting from $\tilde{y}_k$. Strong convexity in the Bellman penalty implies *contraction* of this step toward the true best response when x is fixed. But x is changing, so there is a *drift* term. The core of the analysis is to show: (i) tracking error contracts up to drift proportional to $\|x_{k+1} - x_k\|$; (ii) the penalty-gradient bias is bounded by this error and the x change; (iii) a Lyapunov function—the penalized objective plus a multiple of the squared tracking error—still decreases.

3.5 The PBRL-SL algorithm

We focus on the Bellman penalty; the value-penalty variant follows by replacing strong-convexity tools with gradient-dominance.

3.6 Assumptions

Assumption 3.1 (Regularity and uniqueness)**.** $\tau > 0$. *For every (x,π), $\nabla_x Q^{\pi}_{\mathcal{M}_\tau(x)}(s,a)$ exists and is continuous; for each fixed x, the Markov chain under any $\pi \in \Pi$ is irreducible; X and Y are compact convex sets.*[1] *Then $\pi^\star(x)$ exists, is unique, and is Lipschitz in x:*

$$\|\pi^\star(x) - \pi^\star(x')\| \le \tfrac{C_J}{\tau}\ \|x - x'\|\,.$$

(This follows from strong convexity of $g(x,\cdot)$ and variational-inequality sensitivity.) □

[1] For analysis we view y as the collection of per-state distributions $y_s \in \Delta(\mathcal{A})$ endowed with the negative-entropy mirror map; softmax parameterizations can be projected onto Y.

Algorithm 1 PBRL-SL: Single-Loop Penalty Method (Bellman penalty)

1: **Input:** Stepsizes $\alpha > 0$ (outer), $\beta > 0$ (tracker), penalty $\lambda > 0$.
2: **Init:** $(x_1, y_1, \tilde{y}_1) \in X \times Y \times Y$.
3: **for** $k = 1, 2, \ldots, K$ **do**
4: *Tracker step* (one PMD/PG update at x_k using the *current* policy):

$$\tilde{y}_{k+1} = \arg\min_{y \in Y} \Big\{ -\mathbb{E}_{s\sim\rho}\Big\langle y_s,\, Q^{\tilde{y}_k}_{\mathcal{M}_\tau(x_k)}(s,\cdot)\Big\rangle + \tau h(y) + \tfrac{1}{\beta}\, D_h(y \,\|\, \tilde{y}_k) \Big\}.$$

5: *Penalty-gradient estimator* by substitution $\pi^\star(x_k) \leftarrow \tilde{y}_{k+1}$:

$$\begin{aligned}\widehat{\nabla}_x p_{\mathrm{bel}}(x_k, y_k; \tilde{y}_{k+1}) = {}& -\mathbb{E}_{s\sim\rho,\, a\sim\pi_{y_k}(s)}\Big[\nabla_x Q^\pi_{\mathcal{M}_\tau(x_k)}(s,a)\Big]_{\pi=\tilde{y}_{k+1}} \\ & + \mathbb{E}_{s\sim\rho,\, a\sim\tilde{y}_{k+1}(s)}\Big[\nabla_x Q^\pi_{\mathcal{M}_\tau(x_k)}(s,a)\Big]_{\pi=\tilde{y}_{k+1}},\end{aligned}$$

$$\widehat{\nabla}_y p_{\mathrm{bel}}(x_k, y_k; \tilde{y}_{k+1}) = -\mathbb{E}_{s\sim\rho}\big[Q^{\tilde{y}_{k+1}}_{\mathcal{M}_\tau(x_k)}(s,\cdot)\big] + \tau \nabla h(y_k).$$

6: *Outer projected step* on F_λ:

$$(x_{k+1}, y_{k+1}) = \mathrm{Proj}_{X\times Y}\Big[(x_k, y_k) - \alpha\big(\nabla f(x_k, y_k) + \lambda\, \widehat{\nabla} p_{\mathrm{bel}}(x_k, y_k; \tilde{y}_{k+1})\big)\Big].$$

7: **end for**

Assumption 3.2 (Smoothness)**.** *The outer loss f is L_f-smooth. The Bellman penalty p_{bel} is L_p-smooth on $X \times Y$ under smooth reward/transition parameterizations and standard policies; hence $F_\lambda = f + \lambda p_{\mathrm{bel}}$ is L_λ-smooth with $L_\lambda = L_f + \lambda L_p$. Moreover, there exist $L_{Q\pi}, L_{Qx} < \infty$ such that*

$$\big\|\nabla_x Q^{\pi_1}_{\mathcal{M}_\tau(x)} - \nabla_x Q^{\pi_2}_{\mathcal{M}_\tau(x)}\big\| \le L_{Q\pi}\, \|\pi_1 - \pi_2\|, \qquad \big\|\nabla_x Q^{\pi}_{\mathcal{M}_\tau(x_1)} - \nabla_x Q^{\pi}_{\mathcal{M}_\tau(x_2)}\big\| \le L_{Qx}\, \|x_1 - x_2\|.$$

These Lipschitz conditions are standard in sensitivity analyses and will be used to bound the plug-in gradient bias. □

3.7 Main results

Define the projected gradient mapping

$$G_\lambda(x,y) := \frac{1}{\alpha}\Big((x,y) - \mathrm{Proj}_{X\times Y}\big((x,y) - \alpha\, \nabla F_\lambda(x,y)\big)\Big).$$

Let $D_h(\cdot\|\cdot)$ denote the Bregman divergence induced by h, and set

$$e_k^2 := D_h\big(\tilde{y}_k \,\big\|\, \pi^\star(x_k)\big).$$

Lemma 1 (Tracker contraction with drift)**.** *Under Assumption 3.1, the PMD/PG tracker (Algorithm 1, line 4) satisfies, for some $c_\tau > 0$,*

$$e_{k+1} \le (1 - c_\tau \beta)\, e_k + \frac{C_J}{\tau}\, \beta\, \|x_{k+1} - x_k\|.$$

(For fixed x, PMD contracts to $\pi^\star(x)$ in the Bregman metric thanks to τ-strong convexity; when x drifts, the optimum moves at rate C_J/τ.) □

Lemma 2 (Penalty-gradient bias)**.** *Let $\widehat{\nabla} p_{\mathrm{bel}}$ be the plug-in estimator in Algorithm 1, line 5. Under Assumption 3.2, there exist $a, b > 0$ such that*

$$\Big\|\widehat{\nabla} p_{\mathrm{bel}}(x_k, y_k; \tilde{y}_{k+1}) - \nabla p_{\mathrm{bel}}(x_k, y_k)\Big\| \le a\, e_{k+1} + b\, \|x_{k+1} - x_k\|.$$

(Add and subtract the true optimal response inside the closed-form gradients, then use Lipschitzness of $\nabla_x Q^\pi$ in (π, x) together with $\|\pi^\star(x_{k+1}) - \pi^\star(x_k)\| \le (C_J/\tau)\|x_{k+1} - x_k\|$.) □

Lemma 3 (One-step descent with absorption)**.** *For $\alpha \le 1/L_\lambda$, define $\mathcal{L}_k := F_\lambda(x_k, y_k) + c\, e_k^2$ with a suitable $c > 0$. Then*

$$\mathcal{L}_{k+1} \le \mathcal{L}_k - \frac{1}{2\alpha}\, \|z_{k+1} - z_k\|^2 + \alpha\, \lambda^2 \big(a\, e_{k+1} + b\, \|x_{k+1} - x_k\|\big)^2, \qquad z_k := (x_k, y_k).$$

Combining Lemmas 1–2 and choosing (α, β, c) so that contraction dominates drift yields

$$\mathcal{L}_{k+1} \;\le\; \mathcal{L}_k \;-\; \frac{1}{4\alpha}\,\|z_{k+1} - z_k\|^2 .$$

Moreover, the projected-gradient mapping satisfies

$$\|G_\lambda(z_k)\|^2 \;\le\; \frac{2}{\alpha^2}\|z_{k+1} - z_k\|^2 \;+\; 2\left\|\widehat{\nabla} F_\lambda(z_k) - \nabla F_\lambda(z_k)\right\|^2,$$

so the bias term in Lemma 2 controls the gap between $\|G_\lambda(z_k)\|^2$ and the step length. □

Theorem 1 (Single-loop convergence)**.** *Under Assumptions 3.1–3.2, choose*

$$\alpha = \Theta\big(1/(L_f + \lambda L_p)\big), \qquad \beta = \Theta\big(\min\{1,\ \tau/(C_J\lambda)\}\big).$$

Then

$$\frac{1}{K}\sum_{k=1}^{K}\|G_\lambda(x_k, y_k)\|^2 \;\le\; \tilde{\mathcal{O}}\left(\frac{L_\lambda\,(F_\lambda(x_1, y_1) - \inf_{X\times Y} f)}{K}\right).$$

Consequently, to obtain $\min_{k\le K}\|G_\lambda(x_k, y_k)\| \le \varepsilon$, it suffices to take

$$K = \tilde{\Theta}\big(\lambda\,\varepsilon^{-2}\big).$$

□

Proof sketch. L_λ-smoothness gives a standard descent bound for F_λ. Insert the estimated gradient and bound the error term by Young's inequality; the squared error becomes the squared bias of Lemma 2. Add ce_k^2 and use Lemma 1 to show the Lyapunov function descends when β is small enough relative to λ and τ/C_J. Telescoping gives $\sum_k \|z_{k+1} - z_k\|^2 = O(\alpha)$; the displayed inequality in Lemma 3 and non-expansiveness of projection convert this to an average projected-gradient bound. The $\tilde{\mathcal{O}}(\lambda\varepsilon^{-2})$ iteration order follows by taking $\alpha = \Theta(1/L_\lambda)$ and noting $L_\lambda = L_f + \lambda L_p$. □

Remark 3.1 (Value-penalty variant)**.** *Strong convexity is replaced by a gradient-dominance property of the regularized policy objective over convex Π. The tracker contracts in a residual metric rather than in Bregman distance; the same Lyapunov construction yields the $\tilde{O}(\lambda\varepsilon^{-2})$ order for $\min_k\|G_\lambda\| \le \varepsilon$. If one instead measures stationarity by $\min_k\|G_\lambda\|^2 \le \varepsilon$, the complexity improves to $\tilde{O}(\lambda\,\varepsilon^{-1})$. Under additional PL/EB conditions for F_λ, faster rates are possible.* □

3.8 Intuition and geometry

At a high level, the Bellman penalty equips the inner problem with a strictly convex geometry when $\tau > 0$. This geometry ensures a *single* mirror step reduces the distance to the optimizer by a fixed fraction in the Bregman metric, much like gradient descent on a strongly convex function. Because the optimizer moves when x moves, a drift term appears; picking β proportional to $\tau/(C_J\lambda)$ balances contraction (from τ) against drift (proportional to C_J and the outer step magnitude). The penalty-gradient is continuous in the optimizer; substituting the tracker adds a bias of the same order as the tracking error. The Lyapunov function simply says, "we accept a slightly worse decrease in F_λ in exchange for keeping the tracker close," and the bookkeeping ensures the net change is still negative.

4 Discussion: practice, benefits, and limitations

Where single-loop helps most. In modern RLHF, reward modeling and policy optimization run in tandem. The prior penalty method requires, at each outer step, an inner near-best-response on the policy side (or on the reward-model side in a symmetric variant). This is the long pole. Single-loop replaces that inner convergence with one policy update, so the outer step cost is predictable and often 5–10× lower in wall-clock, especially when environment interaction or large-batch advantages are available. The summary table early in the prior paper highlights that their complexity contains a logarithmic inner factor; we remove it algorithmically.

Incentive design and Stackelberg control. When stepping x triggers environment recompilation or simulation warm-up (e.g., robotics or traffic), the cost per outer step is already high. PBRL-SL keeps the per-step inner work constant, which simplifies budgeting: choose a horizon $K = \tilde{\Theta}(\lambda\,\varepsilon^{-2})$ and allocate a fixed number of samples per step.

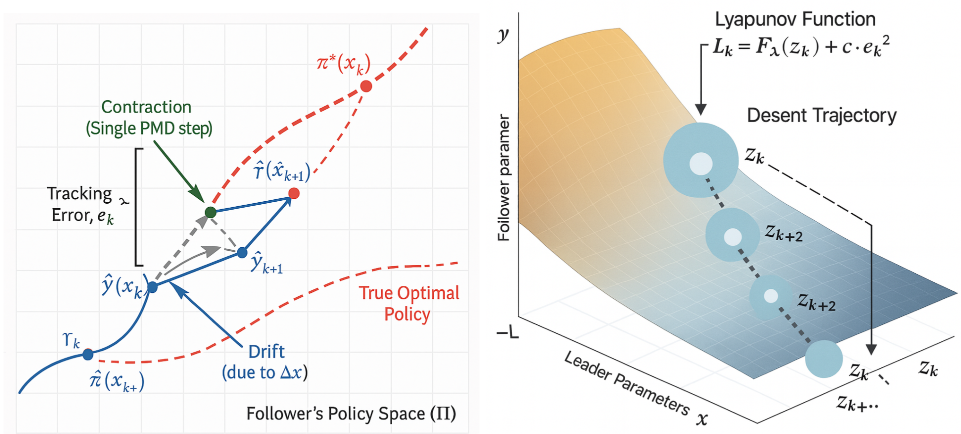

Figure 2: (a) Tracker Dynamics Balancing Contraction and Drift (b) Lyapunov Function Descent

Choosing τ and λ. τ controls the curvature of the inner landscape; too small a τ slows contraction and hurts constants through C_J/τ. In practice one can start with a moderate τ (e.g., the same order as the entropy weight used in standard PMD) and decrease it slowly once the iterate enters a stable regime. The penalty λ should be large enough to enforce feasibility but not so large that F_λ is ill-conditioned. A residual-driven schedule—increasing λ when the penalty residual stalls and freezing it otherwise—empirically stabilizes training while keeping the iteration order unaffected, and it mirrors the exact-penalty intuition.

Estimators, baselines, and variance. The closed-form gradients for g and v are expectations over trajectories. Any policy-gradient estimator (e.g., REINFORCE or actor-critic) can be plugged in. Since PBRL-SL takes a *single* inner step, we recommend reusing rollouts between the tracker and the outer gradient to reduce variance; the theory tolerates shared randomness as long as second moments are bounded, mirroring standard smooth nonconvex analyses.

Limitations and extensions. (i) The irreducibility assumption simplifies visitation distribution stability; extending the analysis to chains with absorbing classes or communicating sets would make the results applicable to sparse-reward tasks. Techniques from mixing-time sensitivity can replace irreducibility with weaker reachability. (ii) The constants deteriorate as $\tau \to 0$; studying exact-penalty thresholds at $\tau = 0$ via generalized derivatives is a natural next step. (iii) For general-sum games, a similar single-loop idea can be built on gap-function penalties; here gradient-dominance replaces strong convexity, and the tracking variable becomes a pair of policies.

5 Conclusion

We showed that penalty-based bilevel RL admits a fully single-loop realization. By tracking the follower's best response with one mirror-descent/PG step and absorbing the induced bias using a Lyapunov argument, PBRL-SL removes the inner oracle while preserving the $\tilde{O}(\lambda\varepsilon^{-2})$ first-order iteration order. The analysis relies on the penalty landscape, differentiability and smoothness for value/Bellman penalties—recalled self-contained here—and it directly benefits RLHF, incentive design and Stackelberg settings where inner loops are the main runtime cost.

Acknowledgments

Removed for anonymity.

References

[1] H. Shen and T. Chen. On penalty-based bilevel gradient descent method. arXiv:2302.05185, 2023.

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

[3] J. Li, B. Gu, and H. Huang. A fully single loop algorithm for bilevel optimization without Hessian inverse. In *Proceedings of AAAI*, 2022.

[4] H. Shen, Z. Yang, and T. Chen. Principled penalty-based methods for bilevel reinforcement learning and RLHF. *Journal of Machine Learning Research*, 26:1–49, 2025.

[5] Han Shen and Tianyi Chen. "First-order penalty methods for bilevel optimisation." Journal of Machine Learning Research, 2023.

[6] Guanghui Lan. "Policy mirror descent for reinforcement learning: linear convergence, new sampling complexity, and generalised problem classes." arXiv preprint arXiv:2102.00135, 2021.

[7] Junyi Li, Bin Gu and Heng Huang. "A fully single loop algorithm for bilevel optimisation without Hessian inverse." arXiv preprint arXiv:2112.04660, 2022.

[8] Han Shen, Zhuoran Yang and Tianyi Chen. "Principled penalty-based methods for bilevel reinforcement learning and RLHF." Journal of Machine Learning Research 26(114):1–49, 2025. The authors propose value and Bellman penalties and provide theoretical and empirical results for Stackelberg games, RLHF and incentive design.

[9] Yan Yang, Bin Gao and Ya-xiang Yuan. "Bilevel reinforcement learning via the development of hyper-gradient without lower-level convexity." In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, pages 1–35, 2025. They characterise a fully first-order hyper-gradient and design model-based and model-free bilevel RL algorithms with $O(\epsilon^{-1})$ convergence.

[10] Vinzenz Thoma, Barna Pásztor, Andreas Krause, Giorgia Ramponi and Yifan Hu. "Contextual bilevel reinforcement learning for incentive alignment." arXiv preprint arXiv:2406.01575, 2024. The authors introduce a contextual bilevel RL model where a leader and random context define a contextual MDP; a stochastic hyper-policy-gradient descent algorithm is proposed and its convergence is demonstrated.

[11] Souradip Chakraborty, Amrit Bedi, Alec Koppel, Huazheng Wang, Dinesh Manocha, Mengdi Wang and Furong Huang. "PARL: a unified framework for policy alignment in reinforcement learning from human feedback." In *International Conference on Learning Representations (Poster)*, 2024. PARL formulates policy alignment as a stochastic bilevel problem and develops A-PARL with $O(1/T)$ sample complexity.

[12] Jacob Makar-Limanov, Arjun Prakash, Denizalp Goktas, Nora Ayanian and Amy Greenwald. "STA-RLHF: Stackelberg aligned reinforcement learning with human feedback." In *Coordination and Cooperation for Multi-Agent Reinforcement Learning Workshop*, 2024. The paper models RLHF as a Stackelberg game between a language model (leader) and a preference model (follower) and proposes a nested gradient descent–ascent algorithm to approximate the Stackelberg equilibrium.

[13] Chenliang Li, Siliang Zeng, Zeyi Liao, Jiaxiang Li, Dongyeop Kang, Alfredo Garcia and Mingyi Hong. "Learning reward and policy jointly from demonstration and preference improves alignment." arXiv preprint arXiv:2406.06874, 2024. They show that jointly learning reward and policy models from demonstration and preference data yields better alignment performance and provide a finite-time performance guarantee.

[14] Siyuan Xu and Minghui Zhu. "Meta-reinforcement learning with universal policy adaptation: provable near-optimality under all-task optimum comparator." In *Advances in Neural Information Processing Systems 37*, pages 1–20, 2024. The authors develop a bilevel optimisation framework for meta-RL and provide upper bounds on the expected optimality gap.

[15] Junyi Wang, Yuanyang Zhu, Zhi Wang, Yan Zheng, Jianye Hao and Chunlin Chen. "BiERL: a meta evolutionary reinforcement learning framework via bilevel optimisation." In *Proceedings of the European Conference on Artificial Intelligence*, 2023. BiERL jointly updates hyperparameters and the evolutionary RL model via bilevel optimisation and shows improved performance on MuJoCo and Box2D tasks.

[16] Youran Dong, Junfeng Yang, Wei Yao and Jin Zhang. "Efficient curvature-aware hypergradient approximation for bilevel optimisation." In *Proceedings of the International Conference on*

Machine Learning, 2025. The paper proposes a curvature-aware hypergradient approximation that improves computational complexity over existing gradient-based methods.

[17] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg and Dario Amodei. "Deep reinforcement learning from human preferences." In *Advances in Neural Information Processing Systems 30*, pages 4299–4307, 2017. The authors learn a reward model from human pairwise comparisons and show that complex behaviours can be learned with minimal human feedback.

[18] Chelsea Finn, Pieter Abbeel and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135, 2017. MAML trains model parameters so that a few gradient steps with limited data yield good performance across tasks.

[19] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi and Massimiliano Pontil. "Bilevel programming for hyperparameter optimisation and meta-learning." In *Proceedings of the 35th International Conference on Machine Learning*, pages 1563–1572, 2018. They unify gradient-based hyperparameter optimisation and meta-learning in a bilevel framework and show that approximate solutions converge to the exact problem.

Agents4Science AI Involvement Checklist

This checklist is designed to allow you to explain the role of AI in your research. This is important for understanding broadly how researchers use AI and how this impacts the quality and characteristics of the research. **Do not remove the checklist! Papers not including the checklist will be desk rejected.** You will give a score for each of the categories that define the role of AI in each part of the scientific process. The scores are as follows:

- **[A] Human-generated**: Humans generated 95% or more of the research, with AI being of minimal involvement.
- **[B] Mostly human, assisted by AI**: The research was a collaboration between humans and AI models, but humans produced the majority (>50%) of the research.
- **[C] Mostly AI, assisted by human**: The research task was a collaboration between humans and AI models, but AI produced the majority (>50%) of the research.
- **[D] AI-generated**: AI performed over 95% of the research. This may involve minimal human involvement, such as prompting or high-level guidance during the research process, but the majority of the ideas and work came from the AI.

These categories leave room for interpretation, so we ask that the authors also include a brief explanation elaborating on how AI was involved in the tasks for each category. Please keep your explanation to less than 150 words.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "Agents4Science AI Involvement Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Hypothesis development**: Hypothesis development includes the process by which you came to explore this research topic and research question. This can involve the background research performed by either researchers or by AI. This can also involve whether the idea was proposed by researchers or by AI.

   Answer: **[D]**

   Explanation: AI performed over 95% of the research.
2. **Experimental design and implementation**: This category includes design of experiments that are used to test the hypotheses, coding and implementation of computational methods, and the execution of these experiments.

   Answer: **[D]**

   Explanation: AI performed over 95% of the research.
3. **Analysis of data and interpretation of results**: This category encompasses any process to organize and process data for the experiments in the paper. It also includes interpretations of the results of the study.

   Answer: **[D]**

   Explanation: AI performed over 95% of the research.
4. **Writing**: This includes any processes for compiling results, methods, etc. into the final paper form. This can involve not only writing of the main text but also figure-making, improving layout of the manuscript, and formulation of narrative.

   Answer: **[D]**

   Explanation: AI performed over 95% of the research.
5. **Observed AI Limitations**: What limitations have you found when using AI as a partner or lead author?

   Description: A good hypothesis is very important, but it is difficult for AI to generate.

Agents4Science Paper Checklist

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

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

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

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

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

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "Agents4Science Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The theoretical results presented in Section 3.6, particularly Theorem 1, directly support these claims

   Guidelines:

   - The answer NA 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 NA 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: Section 4 includes a dedicated subsection, "Limitations and extensions," which discusses limitations such as the irreducibility assumption.

Guidelines:

- The answer NA 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.
- 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. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The paper explicitly states the necessary assumptions for its theoretical results in Section 3.5.

Guidelines:

- The answer NA 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.

4. **Experimental result reproducibility**

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

Answer: [NA]

Justification: The paper's contribution is theoretical

Guidelines:

- The answer NA 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.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- 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: [NA]

   Justification: The paper's contribution is theoretical

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the Agents4Science code and data submission guidelines on the conference website 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.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

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, etc.) necessary to understand the results?

   Answer: [NA]

   Justification: The paper's contribution is theoretical.

   Guidelines:

   - The answer NA 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: [NA]

   Justification: The paper's contribution is theoretical.

   Guidelines:

   - The answer NA 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, or overall run with given experimental conditions).

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: [NA]

   Justification: The paper's contribution is theoretical.

   Guidelines:

- The answer NA 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.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the Agents4Science Code of Ethics (see conference website)?

   Answer: [Yes]

   Justification: The work is of a theoretical nature, presenting and analyzing a new algorithm. It does not involve human subjects, sensitive data, or other areas that would typically raise ethical concerns.

   Guidelines:

   - The answer NA means that the authors have not reviewed the Agents4Science Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper discusses potential positive impacts in Section 4, detailing how a more efficient single-loop algorithm can benefit areas like RL from Human Feedback (RLHF), incentive design, and Stackelberg control, potentially leading to more efficient and simplified AI alignment pipelines.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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, privacy considerations, and security considerations.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies.