Breaking the Bias Barrier in Concave Multi-Objective Reinforcement Learning

Swetha Ganesh[*]
Indian Institute of Technology
Hyderabad, 502284
swethaganesh@ai.iith.ac.in

Jason Chia
Purdue University
West Lafayette, IN 47906
chia6@purdue.edu

Vaneet Aggarwal
Purdue University
West Lafayette, IN 47906
vaneet@purdue.edu

Abstract

While standard reinforcement learning optimizes a single reward signal, many applications require optimizing a nonlinear utility $f(J_1^\pi, \ldots, J_M^\pi)$ over multiple objectives, where each J_m^π denotes the expected discounted return of a distinct reward function. A common approach is concave scalarization, which captures important trade-offs such as fairness and risk sensitivity. However, nonlinear scalarization introduces a fundamental challenge for policy gradient methods: the gradient depends on $\partial f(J^\pi)$, while in practice only empirical return estimates $\hat{J}$ are available. Because f is nonlinear, the plug-in estimator is biased ($\mathbb{E}[\partial f(\hat{J})] \neq \partial f(\mathbb{E}[\hat{J}])$), leading to persistent gradient bias that degrades sample complexity.

In this work we address this bias barrier in concave-scalarized multi-objective reinforcement learning. Prior analyses of empirical-estimator policy-gradient methods in this setting yield $\widetilde{\mathcal{O}}(\epsilon^{-4})$ sample-complexity guarantees. To address this issue, we develop a Natural Policy Gradient (NPG) algorithm equipped with a multi-level Monte Carlo (MLMC) estimator that controls the bias of the scalarization gradient while maintaining low sampling cost. We prove that this approach achieves an $\widetilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity for computing an ϵ-optimal policy up to policy-class approximation error, matching the ϵ-dependence of established bounds for standard single-objective RL. Furthermore, we show that when the scalarization function is second-order smooth, the first-order bias cancels automatically, allowing vanilla NPG to achieve the same $\widetilde{\mathcal{O}}(\epsilon^{-2})$ rate without MLMC. Our results provide improved sample complexity guarantees for concave multi-objective reinforcement learning under policy-gradient methods.

1 Introduction

Standard reinforcement learning (RL) studies how to learn a policy π that maximizes a single scalar objective J^π, defined as the expected discounted return of a reward signal. However, many modern decision-making systems must balance multiple competing objectives rather than optimize a single reward. Examples include trading off throughput and energy consumption in communication systems [5], latency and power usage in queueing and computing systems [6], and efficiency-safety trade-offs in robotic control [23]. In such settings, the performance of a policy is naturally described by a vector of returns $\{J_m^\pi\}_{m=1}^M$, or equivalently $\mathbf{J}^\pi \in \mathbb{R}^M$.

A widely used approach for handling multiple objectives is to introduce a scalarization function and optimize the utility $f(\mathbf{J}^\pi)$, where $f : \mathbb{R}^M \to \mathbb{R}$ is a concave function that captures the trade-offs among objectives. Standard RL is recovered as the special case $M = 1$ with $f(x) = x$. Concave scalarizations are particularly appealing because they naturally encode preferences such as fairness

[*]This work was done while the author was at Purdue University.

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

or risk sensitivity. For example, α-fair utilities widely used in networking, queueing systems, and multi-task learning take the form

$$f(\mathbf{J}^\pi) = \sum_{m=1}^{M} u_\alpha(J_m^\pi), \qquad u_\alpha(x) = \frac{x^{1-\alpha}}{1-\alpha}, \tag{1}$$

where α determines the desired fairness-efficiency trade-off [18, 8].

Despite its conceptual simplicity, optimizing nonlinear utilities in reinforcement learning poses significant theoretical challenges. In particular, value-based methods rely heavily on the additive structure of scalar rewards and Bellman equations, which breaks down under nonlinear scalarization [3, 2]. As a result, policy-gradient methods provide a natural alternative. However, obtaining sharp sample-complexity guarantees for policy-gradient algorithms in this setting has proven to be considerably more difficult [7, 9, 16].

The main challenge stems from estimating the gradient of the scalarized objective. The policy gradient depends on $\partial f(\mathbf{J}^\pi)$ evaluated at the true return vector. In practice, however, $\mathbf{J}^\pi$ is unknown and must be estimated from sampled trajectories. Consequently, policy-gradient estimators rely on plug-in estimates $\widehat{\mathbf{J}}$ and evaluate $\partial f(\widehat{\mathbf{J}})$. Because the scalarization f is nonlinear, this introduces a fundamental bias: even when $\widehat{\mathbf{J}}$ is an unbiased estimate of $\mathbf{J}^\pi$, the gradient estimator satisfies

$$\mathbb{E}[\partial f(\widehat{\mathbf{J}})] \neq \partial f(\mathbb{E}[\widehat{\mathbf{J}}]).$$

This bias persists across iterations and must be controlled through large batch sizes, which substantially increases the sample complexity of policy-gradient methods.

This phenomenon explains the gap in existing theoretical guarantees. The first model-based analysis of concave multi-objective RL was developed in [3, 2], which study the average-reward setting and establish regret guarantees using explicit model estimation. Further, [7] proposed a model-free policy-gradient algorithm and established a sample complexity upper bound of $\widetilde{\mathcal{O}}(\epsilon^{-4})$ samples. This rate is significantly worse than the optimal $\widetilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity known for standard reinforcement learning [21, 10]. The degradation arises precisely from the bias introduced by nonlinear scalarization.

These observations motivate the central question of this work:

> Can we overcome the bias introduced by nonlinear scalarization and obtain an ϵ-optimal policy for concave multi-objective reinforcement learning using only $\widetilde{\mathcal{O}}(\epsilon^{-2})$ samples?

In this paper we answer this question in the affirmative. Our approach combines natural policy gradient (NPG) methods with bias-controlled gradient estimators that mitigate the effects of nonlinear scalarization.

Technical overview and novelty. Our analysis reveals that nonlinear scalarization fundamentally alters the bias-variance trade-off in policy gradient estimation. We show that under standard Lipschitz assumptions the plug-in gradient estimator incurs a bias of order $\mathcal{O}(1/\sqrt{B})$ for batch size B, which forces existing policy-gradient analyses to use $\mathcal{O}(\epsilon^{-2})$ samples per iteration and leads to the overall $\widetilde{\mathcal{O}}(\epsilon^{-4})$ complexity of [7]; even with NPG, this route gives only $\widetilde{\mathcal{O}}(\epsilon^{-3})$ (Appendix J.4).

Our key idea is to explicitly control this bias. We develop a multi-level Monte Carlo (MLMC) estimator that effectively simulates large-batch gradient estimates while requiring only logarithmic expected sampling cost. When combined with natural policy gradient updates, this estimator reduces the scalarization bias sufficiently to recover the order-optimal $\widetilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity. Furthermore, we show that when the scalarization function satisfies a second-order smoothness condition, the leading-order bias cancels automatically through a second-order expansion, enabling vanilla NPG to achieve the same $\widetilde{\mathcal{O}}(\epsilon^{-2})$ rate without MLMC. These results establish improved sample complexity guarantees for policy-gradient methods in concave multi-objective reinforcement learning. Detailed comparison to related work is in Appendix A. The main contributions in this work are summarized as follows.

- **Improved sample complexity via MLMC-NPG.** We develop a natural policy gradient algorithm equipped with a multi-level Monte Carlo (MLMC) estimator that controls the bias arising from nonlinear scalarization. We show that this method achieves $\widetilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity for computing an ϵ-optimal policy, matching the optimal rate known for standard reinforcement learning.

- **Bias cancellation under second-order smooth scalarizations.** We further show that when the scalarization function f satisfies a second-order smoothness condition, the leading-order bias in the plug-in estimator cancels automatically. In this regime, even vanilla NPG achieves $\widetilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity without requiring MLMC.

We complement our theoretical results with empirical evaluations on two MORL environments, Fruit Tree and Breaking Bottles, which illustrate the practical effectiveness of the proposed algorithm (see Appendix B).

2 Problem Setting

We consider an infinite-horizon discounted Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \{r_m\}_{m=1}^M, \gamma, \rho)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $P : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$ is the transition kernel, $r_m : \mathcal{S} \times \mathcal{A} \to [0, 1]$ are the reward functions associated with objectives $m \in [M]$, $\gamma \in (0, 1)$ is the discount factor, and $\rho \in \Delta_{\mathcal{S}}$ is the initial state distribution.

A stationary stochastic policy $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$ maps states to distributions over actions. For each objective $m \in [M]$, the expected discounted return under policy π is

$$J_m^\pi = \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r_m(s_t, a_t)\right]. \tag{2}$$

Similarly, the state-value function $V_m^\pi : \mathcal{S} \to \mathbb{R}$ and state-action value function $Q_m^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ are defined as

$$V_m^\pi(s) = \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r_m(s_t, a_t) \,\middle|\, s_0 = s\right],$$
$$Q_m^\pi(s, a) = \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r_m(s_t, a_t) \,\middle|\, s_0 = s, a_0 = a\right].$$

The advantage function for objective m is $A_m^\pi(s, a) = Q_m^\pi(s, a) - V_m^\pi(s)$. Let $\mathbf{J}^\pi = (J_1^\pi, \ldots, J_M^\pi)^\top \in \mathbb{R}^M$ denote the vector of cumulative returns.

We parameterize the policy as π_θ with $\theta \in \Theta = \mathbb{R}^d$; all parameter updates below are unconstrained. The agent's goal is to solve the optimization problem

$$\max_{\theta \in \Theta} f(\mathbf{J}^{\pi_\theta}), \tag{3}$$

where $f : \mathbb{R}^M \to \mathbb{R}$ is a concave scalarization function.

Assumption 1 (Concavity of f)**.** *The function f is concave. That is, for any random vector X in the domain of f,*

$$f(\mathbb{E}[X]) \geq \mathbb{E}[f(X)].$$

Despite the concavity of f, the optimization problem (3) remains non-concave in the policy parameter θ, since each component $J_m^{\pi_\theta}$ is generally non-concave in θ. We next introduce notation for derivatives of f. For each coordinate $m \in [M]$, let $\partial_m f$ denote the partial derivative of f with respect to its m-th argument. To facilitate convergence analysis we assume a smoothness condition.

Assumption 2 (Smoothness of f)**.** *Let $\mathcal{D} = \left[0, \frac{1}{1-\gamma}\right]^M$ be a compact set containing all return vectors $\{J^{\pi_\theta} : \theta \in \Theta\}$ considered by the algorithm. We assume that f is differentiable on an open neighborhood of $\mathcal{D}$ and that, for each $m \in [M]$, $\partial_m f$ is L_f-Lipschitz on $\mathcal{D}$, i.e.,*

$$|\partial_m f(x) - \partial_m f(y)| \leq L_f \|x - y\|_2, \qquad \forall x, y \in \mathcal{D}.$$

Since rewards lie in $[0, 1]$, every return vector belongs to $\left[0, \frac{1}{1-\gamma}\right]^M$. When f is smooth on the full return cube, we take $\mathcal{D} = \left[0, \frac{1}{1-\gamma}\right]^M$. For utilities with singular derivatives at zero, such as the α-fair

utility in (1), the smoothness assumptions hold for the shifted scalarization

$$f_\delta(J) = \sum_{m=1}^{M} u_\alpha(J_m + \delta),$$

where $\delta > 0$ is fixed and $u_1(x) = \log x$. This scalarization satisfies Assumptions 2 and 6 on the full return cube, with constants depending on δ. By compactness of $\mathcal{D}$ and continuity of $\partial_m f$, there exists $C > 0$ such that $|\partial_m f(y)| \le C$ for all $y \in \mathcal{D}$ and $m \in [M]$.

For $\alpha \ge 0$, one may take

$$C = \delta^{-\alpha}, \qquad L_f = \alpha\delta^{-\alpha-1}, \qquad L_{2,f} = \alpha(\alpha+1)\delta^{-\alpha-2}.$$

These constants may deteriorate as $\delta \downarrow 0$; the stated accuracy rates hold for fixed δ and apply to the shifted objective.

Policy Gradient and Natural Policy Gradient: In this work we adopt a policy gradient approach to solve (3). At iteration k, the parameter θ_k is updated along the gradient $\nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})$. Using the chain rule, we have $\nabla_\theta f(\mathbf{J}^{\pi_\theta}) = \sum_{m=1}^{M} \partial_m f(\mathbf{J}^{\pi_\theta}) \nabla_\theta J_m^{\pi_\theta}$. Applying the policy gradient theorem [26] to each $\nabla_\theta J_m^{\pi_\theta}$ gives

$$\nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}}) = \frac{1}{1-\gamma} \mathbb{E}_{(s,a)\sim\nu_\rho^{\pi_{\theta_k}}} \left[\sum_{m=1}^{M} \partial_m f(\mathbf{J}^{\pi_{\theta_k}}) A_m^{\pi_{\theta_k}}(s,a) \nabla_\theta \log \pi_{\theta_k}(a|s) \right], \tag{4}$$

where $\nu_\rho^{\pi_\theta}(s,a) \coloneqq d_\rho^{\pi_\theta}(s)\pi_\theta(a|s)$ and the discounted state visitation distribution is

$$d_\rho^{\pi_\theta}(s) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s \mid s_0 \sim \rho, \pi_\theta). \tag{5}$$

Note that computing this gradient requires evaluating $\partial_m f(\mathbf{J}^{\pi_\theta})$, which depends on the unknown return vector $\mathbf{J}^{\pi_\theta}$ and must therefore be estimated from sampled trajectories.

Rather than using vanilla policy gradient, we employ the Natural Policy Gradient (NPG) algorithm. At iteration k the parameters are updated as

$$\theta_{k+1} = \theta_k + \alpha\omega_k^*, \tag{6}$$

where $\alpha > 0$ is the step size and $\omega_k^* = F(\theta_k)^\dagger \nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})$ is the NPG direction. Here $F(\theta_k)$ is the Fisher Information Matrix

$$F(\theta_k) = \mathbb{E}_{(s,a)\sim\nu_\rho^{\pi_{\theta_k}}} \left[\nabla_\theta \log \pi_{\theta_k}(a|s) \nabla_\theta \log \pi_{\theta_k}(a|s)^\top \right]. \tag{7}$$

Directly computing the pseudoinverse $F(\theta_k)^\dagger$ can be expensive when the parameter dimension d is large. However, for fixed θ_k, the direction ω_k^* is the solution of

$$\omega_k^* = \arg\min_{\omega\in\mathbb{R}^d} \mathcal{L}_{\nu_\rho^{\pi_{\theta_k}}}(\omega, \theta_k), \text{ where} \tag{8}$$

$$\mathcal{L}_{\nu_\rho^{\pi_{\theta_k}}}(\omega, \theta_k) = \frac{1}{2} \mathbb{E}_{(s,a)\sim\nu_\rho^{\pi_{\theta_k}}} \left[\left(\sum_{m=1}^{M} \partial_m f(\mathbf{J}^{\pi_{\theta_k}}) A_m^{\pi_{\theta_k}}(s,a) - (1-\gamma)\omega^\top \nabla_\theta \log \pi_{\theta_k}(a|s) \right)^2 \right]. \tag{9}$$

The function $\mathcal{L}_{\nu_\rho^{\pi_{\theta_k}}}$ is quadratic in ω with gradient proportional to $F(\theta_k)\omega - \nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})$. Since both the Fisher matrix and the exact gradient are unavailable in closed form, we estimate them from sampled trajectories and apply stochastic gradient descent on $\mathcal{L}_{\nu_\rho^{\pi_{\theta_k}}}$ to obtain an approximate NPG direction ω_k.

3 Algorithm

We develop policy-gradient algorithms for solving the optimization problem $\max_{\theta\in\Theta} f(\mathbf{J}^{\pi_\theta})$. Our approach is based on Natural Policy Gradient (NPG) updates combined with estimators of the policy gradient. The key challenge lies in estimating the scalarization gradient $\partial_m f(\mathbf{J}^{\pi_\theta})$, since the return vector $\mathbf{J}^{\pi_\theta}$ must itself be estimated from sampled trajectories. We first describe the general gradient estimator framework and then present the two estimators used in this work.

Algorithm 1 Natural Policy Gradient for Joint Multi-Objective Optimization

Initialize: Policy parameter θ_0, iteration lengths K, N, trajectory length H, batch sizes B_1, B_2, and step-sizes α, β.
for $k = 0, \dots, K-1$ **do**
  Sample B_1 trajectories of length H $\{\tau_i^H\}_{i=1}^{B_1} \sim \mathbb{P}_{\theta_k}^H$
  Compute $\widehat{\mathbf{J}}_{H,B_1}^{\pi_{\theta_k}}$ using (13)
  Initialize $\omega_0^k \leftarrow 0$
  for $n = 0, \dots, N-1$ **do**
    Sample B_2 trajectories $\{\tau_i^H\}_{i=1}^{B_2}$
    $\hat{g}_n^k \leftarrow \frac{1}{B_2} \sum_{i=1}^{B_2} g(\tau_i^H, \widehat{\mathbf{J}}_{H,B_1}^{\pi_{\theta_k}} | \theta_k)$
    $\widehat{F}_n^k \leftarrow \frac{(1-\gamma)}{B_2} \sum_{i=1}^{B_2} \sum_{t=0}^{H-1} \gamma^t \psi_t^{(i)} (\psi_t^{(i)})^\top$ ▷ $\psi_t^{(i)} := \nabla_\theta \log \pi_\theta(a_t^i | s_t^i)$
    $\omega_{n+1}^k \leftarrow \omega_n^k - \beta(\widehat{F}_n^k \omega_n^k - \hat{g}_n^k)$
  end for
  $\omega_k \leftarrow \omega_N^k$
  $\theta_{k+1} \leftarrow \theta_k + \alpha \omega_k$
end for
Return π_{θ_I} where I is drawn uniformly from $\{0, \dots, K-1\}$ (independent of the algorithm's sampling).

Estimating $\nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})$: Using the policy gradient theorem, the gradient of the scalarized objective can be written as

$$\nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}}) = \mathbb{E}_{\tau \sim \pi_{\theta_k}} \left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi_{\theta_k}(a_t | s_t) \left(\sum_{m=1}^{M} \partial_m f(\mathbf{J}^{\pi_{\theta_k}}) \sum_{h=t}^{\infty} \gamma^h r_m(s_h, a_h) \right) \right]. \tag{10}$$

In practice, trajectories cannot be sampled for infinite horizons. We therefore truncate the horizon at H and define the finite-horizon return vector

$$\mathbf{J}_H^{\pi_\theta} = \left(\mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t r_1(s_t, a_t) \right], \dots, \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t r_M(s_t, a_t) \right] \right)^\top. \tag{11}$$

Based on this truncation, a REINFORCE-style estimator for the gradient is given by

$$g(\tau_i^H, \widehat{\mathbf{J}}_H^{\pi_\theta} | \theta) := \sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(a_t^i | s_t^i) \left(\sum_{m=1}^{M} \partial_m f(\widehat{\mathbf{J}}_H^{\pi_\theta}) \sum_{h=t}^{H-1} \gamma^h r_m(s_h^i, a_h^i) \right), \tag{12}$$

where $\tau_i^H = (s_0^i, a_0^i, \dots, s_{H-1}^i, a_{H-1}^i)$ denotes a sampled trajectory and $\widehat{\mathbf{J}}_H^{\pi_\theta}$ is an estimator of the truncated return vector.

The choice of estimator for $\widehat{\mathbf{J}}_H^{\pi_\theta}$ determines the bias properties of the resulting gradient estimate. We next describe two estimators used in this work.

Empirical Return Estimator (Vanilla NPG): A natural estimator for $\mathbf{J}_H^{\pi_\theta}$ is the empirical average of discounted returns computed from a batch of trajectories. Let $\{\tau_j^H\}_{j=1}^B$ denote B trajectories of length H. The empirical return estimator is defined coordinate-wise as

$$\left[\widehat{\mathbf{J}}_{H,B}^{\pi_\theta} \right]_m = \frac{1}{B} \sum_{j=1}^{B} \sum_{t=0}^{H-1} \gamma^t r_m(s_t^j, a_t^j), \quad m \in [M]. \tag{13}$$

This estimator leads to the algorithm described in Algorithm 1. As we show in the analysis, when the scalarization function satisfies the second-order smoothness condition introduced in Assumption 6, the resulting gradient bias decays at rate $O(B^{-1})$. This rate is sufficient for vanilla Natural Policy Gradient to achieve an improved sample complexity.

Algorithm 2 MLMC Natural Policy Gradient

Initialize: Policy parameter θ_0, iteration lengths K, N, trajectory length H, batch threshold $B_{\max}$, batch size B, and step-sizes α, β.
for $k = 0, \dots, K-1$ **do**
  Draw $Q_k \sim \mathrm{Geom}(1/2)$
  $\bar{Q}_k \leftarrow Q_k \mathbf{1}_{\{2^{Q_k} \le B_{\max}\}}$
  Sample $B_k = 2^{\bar{Q}_k}$ trajectories of length H $\{\tau_i^H\}_{i=1}^{B_k} \sim \mathbb{P}_{\theta_k}^H$
  Compute $\partial_m f(\widehat{\mathbf{J}}_{H,\mathrm{MLMC}}^{\pi_{\theta_k}})$ using (14)
  Initialize $\omega_0^k \leftarrow 0$
  for $n = 0, \dots, N-1$ **do**
    Sample B trajectories
    $\hat{g}_n^k \leftarrow \frac{1}{B} \sum_{i=1}^{B} g(\tau_i^H, \widehat{\mathbf{J}}_{H,\mathrm{MLMC}}^{\pi_{\theta_k}} | \theta_k)$
    $\widehat{F}_n^k \leftarrow \frac{(1-\gamma)}{B} \sum_{i=1}^{B} \sum_{t=0}^{H-1} \gamma^t \psi_t^{(i)} (\psi_t^{(i)})^\top$ ▷ $\psi_t^{(i)} := \nabla_\theta \log \pi_\theta(a_t^i | s_t^i)$
    $\omega_{n+1}^k \leftarrow \omega_n^k - \beta(\widehat{F}_n^k \omega_n^k - \hat{g}_n^k)$
  end for
  $\omega_k \leftarrow \omega_N^k$
  $\theta_{k+1} \leftarrow \theta_k + \alpha \omega_k$
end for
Return π_{θ_I} where I is drawn uniformly from $\{0, \dots, K-1\}$ (independent of the algorithm's sampling).

MLMC Return Estimator: When only Lipschitz continuity of $\partial_m f$ is assumed, the empirical estimator leads to a gradient bias of order $O(B^{-1/2})$. Achieving sufficiently small bias would therefore require large batch sizes at every iteration, leading to suboptimal sample complexity. To overcome this limitation, we introduce a truncated multi-level Monte Carlo (MLMC) estimator that simulates large-batch gradients while requiring only logarithmic expected sampling cost.

Draw a random level $Q \sim \mathrm{Geom}(1/2)$ with $\Pr(Q = q) = 2^{-q}$ for $q = 1, 2, \dots$. Let $\bar{Q} = \mathbf{1}_{\{2^Q \le B_{\max}\}} Q$ denote the truncated level and sample $2^{\bar{Q}}$ trajectories. For each coordinate $m \in [M]$ we define

$$\partial_m f(\widehat{\mathbf{J}}_{H,\mathrm{MLMC}}^{\pi_\theta}) = \partial_m f(\widehat{\mathbf{J}}_{H,1}^{\pi_\theta}) + \mathbf{1}_{\{2^Q \le B_{\max}\}} 2^Q \left(\partial_m f(\widehat{\mathbf{J}}_{H,2^Q}^{\pi_\theta}) - \partial_m f(\widehat{\mathbf{J}}_{H,2^{Q-1}}^{\pi_\theta}) \right). \tag{14}$$

The coarse estimate uses the first half of the fine batch, and the base estimate uses its first trajectory. When $2^Q > B_{\max}$, the correction in (14) vanishes and only the base trajectory is sampled. Equation (14) estimates the scalarization derivatives used as weights in (12). Note that $\partial_m f(\widehat{\mathbf{J}}_{H,\mathrm{MLMC}}^{\pi_\theta})$ denotes an estimator of $\partial_m f(\mathbf{J}_H^{\pi_\theta})$, not $\partial_m f$ evaluated at a point; with slight abuse of notation, $g(\tau_i^H, \widehat{\mathbf{J}}_{H,\mathrm{MLMC}}^{\pi_\theta} | \theta)$ denotes (12) with these estimates used as the weights. The resulting algorithm is shown in Algorithm 2.

Natural Policy Gradient Estimation and Policy Update: Both algorithms compute the natural policy gradient direction by approximately solving the quadratic objective introduced in (9). The Fisher information matrix can be estimated from a trajectory of length H as

$$\widehat{F}(\theta_k) = (1-\gamma) \sum_{t=0}^{H-1} \gamma^t \nabla_\theta \log \pi_{\theta_k}(a_t | s_t) \nabla_\theta \log \pi_{\theta_k}(a_t | s_t)^\top. \tag{15}$$

Given a gradient estimate $\widehat{g}_k \approx \nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})$, we perform stochastic gradient descent on the quadratic objective: $\omega_{n+1}^k = \omega_n^k - \beta \left(\widehat{F}(\theta_k) \omega_n^k - \widehat{g}_k \right)$. After N iterations we set $\omega_k = \omega_N^k$ and update the policy parameters according to $\theta_{k+1} = \theta_k + \alpha\, \omega_k$. Algorithm 1 and Algorithm 2 correspond to the vanilla and MLMC gradient estimators, respectively.

We note that, conditional on the current policy parameter θ_k, the trajectories used to estimate the scalarization gradient are independent of the trajectories sampled in the NPG inner loop. The scalarization-gradient estimate is held fixed during that inner loop, and each inner iteration draws a fresh trajectory batch.

4 Main Results

We now present the convergence guarantees for the proposed algorithms. We first state the conditions used in the analysis.

Assumption 3. *The score function $\nabla_\theta \log \pi_\theta(a|s)$ is uniformly bounded and Lipschitz. Specifically, for all $\theta, \theta_1, \theta_2 \in \Theta$ and $(s,a) \in \mathcal{S} \times \mathcal{A}$,*

$$\|\nabla_\theta \log \pi_\theta(a|s)\| \le G_1,$$
$$\|\nabla_\theta \log \pi_{\theta_1}(a|s) - \nabla_\theta \log \pi_{\theta_2}(a|s)\| \le G_2 \|\theta_1 - \theta_2\|.$$

Assumption 4. *For all $\theta \in \mathbb{R}^d$, the Fisher information matrix induced by policy π_θ and initial distribution ρ satisfies*

$$F_\rho(\theta) := \mathbb{E}_{s\sim d_\rho^{\pi_\theta}} \mathbb{E}_{a\sim\pi_\theta(\cdot|s)} \left[\nabla_\theta \log \pi_\theta(a|s) \nabla_\theta \log \pi_\theta(a|s)^\top\right] \succeq \mu I$$

for some $\mu > 0$.

Assumption 5. *For all $\theta \in \Theta$,*

$$\mathcal{L}_{\nu_\rho^{\pi^*}}(\omega_\theta^*, \theta) \le \varepsilon_{\text{bias}},$$

where π^* denotes an optimal policy, $\nu_\rho^{\pi^*}(s,a) := d_\rho^{\pi^*}(s)\pi^*(a|s)$, $\mathcal{L}_{\nu_\rho^{\pi^*}}$ is the loss in (9) with the expectation taken over $\nu_\rho^{\pi^*}$ instead of $\nu_\rho^{\pi_\theta}$, and $\omega_\theta^* = F(\theta)^{-1}\nabla_\theta f(\mathbf{J}^{\pi_\theta})$ denotes the natural policy gradient direction at θ.

Assumption 6 (Second-Order Smoothness)**.** *For each $m \in [M]$, the partial derivative $\partial_m f$ is $L_{2,f}$-smooth on $\left[0, \frac{1}{1-\gamma}\right]^M$, i.e.,*

$$\|\nabla \partial_m f(x) - \nabla \partial_m f(y)\| \le L_{2,f}\|x - y\|, \qquad \forall x, y \in \left[0, \tfrac{1}{1-\gamma}\right]^M.$$

Assumptions 3–5 follow a line of global convergence analyses for parameterized policy-gradient methods [19, 10, 21, 20, 25, 12, 13, 11]. Assumption 3 requires uniformly bounded and Lipschitz score functions, while Assumption 4 imposes uniform Fisher non-degeneracy. These are substantive restrictions on the policy class. In particular, standard redundant softmax logits yield a singular Fisher matrix; even after removing redundancy, Fisher eigenvalues can approach zero as action probabilities vanish. Expressivity and Fisher non-degeneracy must therefore be checked separately. Assumption 5 quantifies the transferred approximation error, which is retained in our guarantees [1]. The results are conditional on these assumptions holding jointly. Assumption 6 imposes additional regularity on the scalarization gradient and is used only for our second result.

We consider each NPG inner loop initialized at $\omega_0^k = 0$ and use

$$\beta = \frac{\mu}{4G_1^4}, \quad N = \left\lceil \frac{32G_1^4}{\mu^2}\log(1/\epsilon)\right\rceil, \quad H = \left\lceil \frac{\log\big(\max\{\epsilon^{-2}, 2G_1^2/\mu\}\big)}{\log(1/\gamma)}\right\rceil.$$

Further, in both theorems, we set $K = \lceil 1/(\alpha\epsilon)\rceil$. Assume the initial expected KL divergence from π^* to π_{θ_0} is finite. All statements hold with problem parameters and the initial policy fixed. Here L_J denotes the smoothness constant of $\theta \mapsto f(\mathbf{J}^{\pi_\theta})$ (Lemma 9 in Appendix D),

$$L_J = MC\left(\frac{G_2}{(1-\gamma)^2} + \frac{(1+\gamma)G_1^2}{(1-\gamma)^3}\right) + \frac{M^{3/2}L_f G_1^2}{(1-\gamma)^4},$$

where C is the bound on $|\partial_m f|$ over $\mathcal{D}$ introduced after Assumption 2.

Theorem 1 (MLMC-NPG)**.** *Let Assumptions 1–5 hold and use the common choices of H, N, β, and initialization stated above. Set*

$$\alpha = \min\left\{1, \frac{\mu^2}{4L_J G_1^2}\right\}\frac{\epsilon}{1+\log(1/\epsilon)}, \qquad B_{\max} = \epsilon^{-2}, \qquad B = 1.$$

Then Algorithm 2 satisfies

$$f(\mathbf{J}^{\pi^*}) - \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[f(\mathbf{J}^{\pi_{\theta_k}})] \le \frac{\sqrt{2\varepsilon_{\text{bias}}}}{1-\gamma} + \mathcal{O}(\epsilon). \tag{16}$$

The same bound holds for its uniformly selected output policy in expectation. Both the expected trajectory count and expected transition count are $\widetilde{\mathcal{O}}(\epsilon^{-2})$.

The proof of Theorem 1 is in Appendix J.2. Our second result shows that MLMC is not necessary when the scalarization admits additional second-order smoothness.

Theorem 2 (Vanilla NPG under Second-Order Smoothness)**.** *Let Assumptions 1–6 hold and use the common choices of H, N, β, and initialization stated above. Set*

$$\alpha = \min\left\{1, \frac{\mu^2}{4L_J G_1^2}\right\}, \qquad B_1 = B_2 = \left\lceil \frac{1}{\epsilon(1-\gamma)^2} \right\rceil.$$

Then Algorithm 1 satisfies

$$f(\mathbf{J}^{\pi^*}) - \frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}[f(\mathbf{J}^{\pi_{\theta_k}})] \le \frac{\sqrt{2\varepsilon_{\text{bias}}}}{1-\gamma} + \mathcal{O}(\epsilon). \tag{17}$$

The same bound holds for its uniformly selected output policy in expectation. Both the trajectory count and transition count are $\widetilde{\mathcal{O}}(\epsilon^{-2})$.

The proof of Theorem 2 is in Appendix J.3. We note that Theorem 1 also covers scalarizations that do not satisfy Assumption 6. For example, let $U = (1-\gamma)^{-1}$ and define

$$f(x) = \sum_{m=1}^{M}\left[x_m - \frac{1}{2U}\left(x_m - \frac{U}{2}\right)_+^2\right], \qquad x \in [0,U]^M,$$

where $z_+ = \max\{z, 0\}$. This utility is concave and increasing, and its partial derivatives are Lipschitz. However, these derivatives have kinks at $x_m = U/2$, so Assumption 6 fails. Thus, the MLMC guarantee applies to scalarizations beyond the stronger smoothness regime required by Theorem 2.

5 Convergence Analysis of Bias-Controlled Policy Gradient (Proof Sketch)

Our analysis separates the optimization error of natural policy gradient from the statistical error introduced by gradient estimation. We first derive a general convergence bound for inexact NPG updates in terms of the bias and variance of the estimated update direction.

5.1 A General NPG Framework

We begin with a general convergence bound for approximate natural policy gradient updates.

Lemma 1 (General Framework)**.** *Suppose the parameter update satisfies $\theta_{k+1} = \theta_k + \alpha\omega_k$. Under Assumptions 1, 2, 3, and 5,*

$$\begin{aligned} f(\mathbf{J}^{\pi^*}) - \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[f(\mathbf{J}^{\pi_{\theta_k}})] \le & \frac{\sqrt{2\varepsilon_{\text{bias}}}}{1-\gamma} + \frac{G_1}{K}\sum_{k=0}^{K-1}\mathbb{E}\,\|\mathbb{E}[\omega_k|\theta_k] - \omega_k^*\| \\ & + \frac{G_2\alpha}{2K}\sum_{k=0}^{K-1}\mathbb{E}[\|\omega_k\|^2] + \frac{1}{\alpha K}\mathbb{E}_{s\sim d_\rho^{\pi^*}}\left[\mathrm{KL}(\pi^*(\cdot|s)\|\pi_{\theta_0}(\cdot|s))\right], \end{aligned} \tag{18}$$

where $\omega_k^* := \omega_{\theta_k}^*$.

A related bound appears in [7], but is expressed in terms of $\mathbb{E}\,\|\omega_k - \omega_k^*\|$. Our bound instead depends on $\|\,\mathbb{E}[\omega_k] - \omega_k^*\|$, which is sharper for our bias-sensitive analysis. The proof is in Appendix C.

To apply Lemma 1, it remains to bound the estimation error of the update direction. Since ω_k is computed by running stochastic gradient descent on the quadratic objective defining the natural gradient direction, we obtain the following generic control.

Lemma 2 (NPG Estimation Errors)**.** *Let Assumptions 3 and 4 hold. Then the estimated update direction ω_k satisfies*

$$\begin{aligned} \mathbb{E}\left[\|\omega_k - \omega_k^*\|^2\right] &\le \mathcal{O}\left(e^{-cN}R_0^2 + \gamma^H + \frac{1}{B} + \sigma_g^2\right), \\ \|\mathbb{E}[\omega_k|\theta_k] - \omega_k^*\|^2 &\le \mathcal{O}\left(e^{-cN}R_0^2 + \gamma^H + \bar{\delta}_g^2\right), \end{aligned}$$

where

$$\bar{\delta}_g = \left\|\mathbb{E}[\widehat{g}_n^k \mid \theta_k] - \nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})\right\|, \qquad \sigma_g^2 = \mathbb{E}\left[\left\|\widehat{g}_n^k - \nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})\right\|^2 \,\middle|\, \theta_k\right].$$

Here $R_0 = \|\omega_0^k - \omega_k^*\|$ is the initial inner-loop error, $c = \mu^2/(16G_1^4)$, and B is the inner-loop batch size (B_2 in Algorithm 1).

Lemma 2, proved in Appendix E, reduces the analysis to controlling the bias and second moment of the policy-gradient estimator.

5.2 Bias and Variance of the Gradient Estimator

We now characterize the two sources of gradient estimation error. The first is truncation: from Lemma 9,

$$\left\|\mathbb{E}[g(\tau_i^H, \mathbf{J}_H^{\pi_\theta} \mid \theta)] - \nabla_\theta f(\mathbf{J}^{\pi_\theta})\right\| \le \mathcal{O}((1+H)\gamma^H). \tag{19}$$

The second, and more subtle, source is the plug-in error caused by estimating the return vector. Unlike standard RL, the scalarization is nonlinear, so in general

$$\mathbb{E}[\partial_m f(\widehat{\mathbf{J}}_H^{\pi_\theta})] \neq \partial_m f(\mathbb{E}[\widehat{\mathbf{J}}_H^{\pi_\theta}]).$$

This non-commutativity creates a persistent bias in the gradient estimator.

Lemma 3. *Let Assumption 3 hold. Then:*

$$\textit{(i)} \quad \left\|\nabla_\theta f(\mathbf{J}^{\pi_\theta}) - \mathbb{E}[g(\tau_i^H, \widehat{\mathbf{J}}_H^{\pi_\theta} \mid \theta)]\right\| \le \mathcal{O}\left(\frac{G_1}{(1-\gamma)^2}\sum_{m=1}^{M}\left|\partial_m f(\mathbf{J}_H^{\pi_\theta}) - \mathbb{E}\left[\partial_m f(\widehat{\mathbf{J}}_H^{\pi_\theta})\right]\right|\right) + \mathcal{O}((1+H)\gamma^H) \tag{20}$$

$$\textit{(ii)} \quad \mathbb{E}\left\|g(\tau_i^H, \widehat{\mathbf{J}}_H^{\pi_\theta} \mid \theta) - g(\tau_i^H, \mathbf{J}_H^{\pi_\theta} \mid \theta)\right\|^2 \le \frac{G_1^2 M}{(1-\gamma)^4}\sum_{m=1}^{M}\mathbb{E}\left|\partial_m f(\widehat{\mathbf{J}}_H^{\pi_\theta}) - \partial_m f(\mathbf{J}_H^{\pi_\theta})\right|^2. \tag{21}$$

Lemma 3 is proved in Appendix F. If $\partial_m f$ is merely Lipschitz, then

$$\left|\partial_m f(\mathbf{J}_H^{\pi_\theta}) - \mathbb{E}\left[\partial_m f(\widehat{\mathbf{J}}_H^{\pi_\theta})\right]\right| \le L_f \mathbb{E}\left\|\mathbf{J}_H^{\pi_\theta} - \widehat{\mathbf{J}}_H^{\pi_\theta}\right\|. \tag{22}$$

Thus the gradient bias is governed by the accuracy of the return estimator.

5.3 The Bias Barrier for the Empirical Estimator

Let $\widehat{\mathbf{J}}_{H,B}^{\pi_\theta}$ denote the empirical return estimate formed from a batch of size B.

Lemma 4. *The empirical return estimator satisfies*

$$\mathbb{E}\left\|\widehat{\mathbf{J}}_{H,B}^{\pi_\theta} - \mathbf{J}_H^{\pi_\theta}\right\|^2 \le \frac{M}{(1-\gamma)^2 B}. \tag{23}$$

Lemma 4 is proved in Appendix G. Combining Lemmas 3 and 4, we obtain

$$\left\|\nabla_\theta f(\mathbf{J}^{\pi_\theta}) - \mathbb{E}\left[g(\tau_i^H, \widehat{\mathbf{J}}_{H,B}^{\pi_\theta} \mid \theta)\right]\right\| \le \mathcal{O}\left(\frac{G_1 M^{3/2} L_f}{(1-\gamma)^3\sqrt{B}} + (1+H)\gamma^H\right). \tag{24}$$

Equation (24) identifies a central bottleneck in empirical-estimator policy-gradient methods: under Lipschitz continuity alone, the scalarization-induced bias is controlled at rate $\mathcal{O}(B^{-1/2})$. Achieving an $\mathcal{O}(\epsilon)$ bias guarantee through this bound entails $B = \Theta(\epsilon^{-2})$ trajectories per policy iteration. This large-batch sampling cost is a key source of the $\widetilde{O}(\epsilon^{-4})$ sample-complexity guarantee in [7]. Even when combined with NPG, this route yields only an $\widetilde{\mathcal{O}}(\epsilon^{-3})$ guarantee under Assumptions 1–5 (Appendix J.4). Our approach overcomes this bottleneck by attaining the bias of a batch-$B_{\max}$ empirical estimator using only $\mathcal{O}(\log B_{\max})$ trajectories in expectation. Combined with NPG and control of the resulting variance, this yields the $\widetilde{\mathcal{O}}(\epsilon^{-2})$ guarantee of Theorem 1.

5.4 MLMC Estimator

We first show that the above barrier can be overcome without Assumption 6 by using a multi-level Monte Carlo estimator.

Lemma 5. *Consider the MLMC estimator $g(\tau_i^H, \widehat{\mathbf{J}}^{\pi_\theta}_{H,\mathrm{MLMC}} \mid \theta)$, truncated at $B_{\max}$. Under Assumptions 1–3,*

$$\textit{(i)} \quad \mathbb{E}[B_k] = \mathcal{O}(\log B_{\max}),$$

$$\textit{(ii)} \quad \left\| \nabla_\theta f(\mathbf{J}^{\pi_\theta}) - \mathbb{E}\left[g(\tau_i^H, \widehat{\mathbf{J}}^{\pi_\theta}_{H,\mathrm{MLMC}} \mid \theta) \right] \right\| \le \mathcal{O}\left(\frac{G_1 M^{3/2} L_f}{(1-\gamma)^3 \sqrt{B_{\max}}} + (1+H)\gamma^H \right),$$

$$\textit{(iii)} \quad \mathbb{E}\left\| \nabla_\theta f(\mathbf{J}^{\pi_\theta}) - g(\tau_i^H, \widehat{\mathbf{J}}^{\pi_\theta}_{H,\mathrm{MLMC}} \mid \theta) \right\|^2 \le \frac{6C^2 G_1^2 M^2}{(1-\gamma)^4} + \frac{24 L_f^2 G_1^2 M^3 \lfloor \log_2 B_{\max} \rfloor}{(1-\gamma)^6}.$$

The key point is that the MLMC estimator telescopes to the same bias level as a large-batch empirical estimator with batch size $B_{\max}$, while requiring only logarithmic expected sampling cost. The proof is in Appendix H.

5.5 Improved Bias under Second-Order Smoothness

We next show that the same barrier disappears for the empirical estimator when the scalarization is second-order smooth.

Lemma 6. *Consider the empirical estimator $g(\tau_i^H, \widehat{\mathbf{J}}^{\pi_\theta}_{H,B} \mid \theta)$. Under Assumptions 1–3 and 6,*

$$\textit{(i)} \quad \left\| \nabla_\theta f(\mathbf{J}^{\pi_\theta}) - \mathbb{E}\left[g(\tau_i^H, \widehat{\mathbf{J}}^{\pi_\theta}_{H,B} \mid \theta) \right] \right\| \le \mathcal{O}\left(\frac{L_{2,f} G_1 M^2}{(1-\gamma)^4 B} + (1+H)\gamma^H \right).$$

(ii) For the averaged gradient estimator $\widehat{g}_n^k$ defined in Algorithm 1, using independent return and gradient batches of sizes B_1 and B_2, respectively,

$$\mathbb{E}\left[\left\| \widehat{g}_n^k - \nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}}) \right\|^2 \,\middle|\, \theta_k \right] \le \frac{C^2 G_1^2 M^2}{(1-\gamma)^4 B_2} + \frac{2 L_f^2 G_1^2 M^3}{(1-\gamma)^6 B_1} + \mathcal{O}\big((1+H)^2 \gamma^{2H}\big).$$

The constant in the final term depends only on fixed problem parameters.

Lemma 6 shows that second-order smoothness improves the plug-in bias from $O(B^{-1/2})$ to $O(B^{-1})$. Hence the empirical estimator itself becomes sufficient for order-optimal natural policy gradient. The proof is in Appendix I.

Together, Lemmas 5 and 6 identify two distinct mechanisms for overcoming the bias barrier in concave-scalarized multi-objective reinforcement learning. Substituting these bounds into Lemma 1 and the update-direction error bounds of Lemma 2 yields Theorems 1 and 2 (Appendix J).

6 Conclusion

We study *concave-scalarized multi-objective reinforcement learning* in a policy-gradient framework and establish improved sample complexity guarantees for this setting. A key challenge arises from the nonlinearity of the scalarization function: estimating the return vector from trajectories introduces bias in the policy-gradient estimator, which fundamentally limits standard analyses. We show that this barrier can be overcome by controlling the bias of the gradient estimator. In particular, Natural Policy Gradient (NPG) combined with a multi-level Monte Carlo (MLMC) estimator achieves an $\widetilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity, matching the ϵ-dependence known for standard single-objective RL, without the second-order smoothness condition (Assumption 6). Moreover, when the partial derivatives of the scalarization have Lipschitz-continuous gradients (Assumption 6), we show that vanilla NPG attains the same rate through a refined bias analysis.

While this problem is closely related to reinforcement learning with general utilities (RLGU) [32], and the two formulations are equivalent in finite state-action spaces, they differ in the presence of function approximation. Whether our results extend to the RLGU setting with policy parameterization remains an open problem.

References

[1] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *The Journal of Machine Learning Research*, 22(1):4431–4506, 2021.

[2] M. Agarwal and V. Aggarwal. Reinforcement learning for joint optimization of multiple rewards. *Journal of Machine Learning Research*, 24(49):1–41, 2023.

[3] M. Agarwal, V. Aggarwal, and T. Lan. Multi-objective reinforcement learning with non-linear scalarization. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pages 9–17, 2022.

[4] M. Agarwal, Q. Bai, and V. Aggarwal. Concave utility reinforcement learning with zero-constraint violations. *Transactions on Machine Learning Research*, 2022.

[5] V. Aggarwal, M. R. Bell, A. Elgabli, X. Wang, and S. Zhong. Joint energy-bandwidth allocation for multiuser channels with cooperating hybrid energy nodes. *IEEE Transactions on Vehicular Technology*, 66:9880–9889, 2017.

[6] A. Badita, P. Parag, and V. Aggarwal. Optimal server selection for straggler mitigation. *IEEE/ACM Trans. Netw.*, 28(2):709–721, Apr. 2020. ISSN 1063-6692. doi: 10.1109/TNET.2020.2973224.

[7] Q. Bai, M. Agarwal, and V. Aggarwal. Joint optimization of concave scalarized multi-objective reinforcement learning with policy gradient based algorithm. *Journal of Artificial Intelligence Research*, 74, 2022. ISSN 1076-9757. doi: 10.1613/jair.1.13981.

[8] H. Ban and K. Ji. Fair resource allocation in multi-task learning. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=KLmWRMg6nL`.

[9] A. Barakat, S. Chakraborty, P. Yu, P. Tokekar, and A. S. Bedi. On the global optimality of policy gradient methods in general utility reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=aq9Nc5NvNc`.

[10] I. Fatkhullin, A. Barakat, A. Kireeva, and N. He. Stochastic policy gradient methods: Improved sample complexity for fisher-non-degenerate policies. In *International Conference on Machine Learning*, pages 9827–9869. PMLR, 2023.

[11] S. Ganesh and V. Aggarwal. Regret analysis of average-reward unichain MDPs via an actor-critic approach. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=rl2aAyJfQz`.

[12] S. Ganesh, W. U. Mondal, and V. Aggarwal. A sharper global convergence analysis for average reward reinforcement learning via an actor-critic approach. In *The 42nd International Conference on Machine Learning (ICML)*, 2025.

[13] S. Ganesh, W. U. Mondal, and V. Aggarwal. Order-optimal regret with novel policy gradient approaches in infinite horizon average reward MDPs. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.

[14] M. Geist, J. Pérolat, M. Laurière, R. Elie, S. Perrin, O. Bachem, R. Munos, and O. Pietquin. Concave utility reinforcement learning: The mean-field game viewpoint. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, page 489–497, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450392136.

[15] E. Hazan, S. Kakade, K. Singh, and A. Van Soest. Provably efficient maximum entropy exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2681–2691. PMLR, 09–15 Jun 2019.

[16] A. Huang and N. Jiang. Occupancy-based policy gradient: Estimation, convergence, and optimality. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 416–468. Curran Associates, Inc., 2024. doi: 10.52202/079017-0014.

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017. URL `https://arxiv.org/abs/1412.6980`.

[18] T. Lan, D. Kao, M. Chiang, and A. Sabharwal. An axiomatic theory of fairness in network resource allocation. In *2010 Proceedings IEEE INFOCOM*, pages 1–9, 2010. doi: 10.1109/INFCOM.2010.5461911.

[19] Y. Liu, K. Zhang, T. Basar, and W. Yin. An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods. *Advances in Neural Information Processing Systems*, 33:7624–7636, 2020.

[20] S. Masiha, S. Salehkaleybar, N. He, N. Kiyavash, and P. Thiran. Stochastic second-order methods provably beat sgd for gradient-dominated functions. *arXiv preprint arXiv:2205.12856v1*, 2022.

[21] W. U. Mondal and V. Aggarwal. Improved sample complexity analysis of natural policy gradient algorithm with general parameterization for infinite horizon discounted reward markov decision processes. In *International Conference on Artificial Intelligence and Statistics*, pages 3097–3105. PMLR, 2024.

[22] M. Mutti, R. D. Santi, P. D. Bartolomeis, and M. Restelli. Convex reinforcement learning in finite trials. *Journal of Machine Learning Research*, 24(250):1–42, 2023. URL `http://jmlr.org/papers/v24/22-1514.html`.

[23] M. Nishimura and R. Yonetani. L2b: Learning to balance the safety-efficiency trade-off in interactive crowd-aware robot navigation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 11004–11010. IEEE Press, 2020. doi: 10.1109/IROS45743.2020.9341519. URL `https://doi.org/10.1109/IROS45743.2020.9341519`.

[24] N. Peng, M. Tian, and B. Fain. Multi-objective reinforcement learning with nonlinear preferences: Provable approximation for maximizing expected scalarized return. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, page 1632–1640. International Foundation for Autonomous Agents and Multiagent Systems, 2025. ISBN 9798400714269.

[25] W. A. Suttle, A. Bedi, B. Patel, B. M. Sadler, A. Koppel, and D. Manocha. Beyond exponentially fast mixing in average-reward reinforcement learning via multi-level Monte Carlo actor-critic. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 33240–33267. PMLR, 23–29 Jul 2023.

[26] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[27] P. Vamplew, C. Foale, R. Dazeley, and A. Bignold. Potential-based multiobjective reinforcement learning approaches to low-impact agents for ai safety. *Engineering Applications of Artificial Intelligence*, 100:104186, 2021. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2021.104186. URL `https://www.sciencedirect.com/science/article/pii/S0952197621000336`.

[28] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992. ISSN 1573-0565. doi: 10.1007/BF00992696. URL `https://doi.org/10.1007/BF00992696`.

[29] Y. Xu, S. Ganesh, W. Mondal, Q. Bai, and V. Aggarwal. Global convergence for average reward constrained mdps with primal-dual actor critic algorithm. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 92592–92627. Curran Associates, Inc., 2025. doi: 10.52202/085713-3099.

[30] R. Yang, X. Sun, and K. Narasimhan. *A generalized algorithm for multi-objective reinforcement learning and policy adaptation*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[31] T. Zahavy, B. O' Donoghue, G. Desjardins, and S. Singh. Reward is enough for convex mdps. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 25746–25759. Curran Associates, Inc., 2021.

[32] J. Zhang, A. Koppel, A. S. Bedi, C. Szepesvari, and M. Wang. Variational policy gradient method for reinforcement learning with general utilities. In *Advances in Neural Information Processing Systems*, volume 33, pages 4572–4583. Curran Associates, Inc., 2020.

[33] J. Zhang, C. Ni, Z. Yu, C. Szepesvari, and M. Wang. On the convergence and sample efficiency of variance-reduced policy gradient method. In *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=Re_VXFOyyO`.

[34] R. Zhou, T. Liu, D. Kalathil, P. Kumar, and C. Tian. Anchor-changing regularized natural policy gradient for multi-objective reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=kY1RbKE7DWE`.

A Related Works

Concave multi-objective reinforcement learning. The formulation of reinforcement learning with concave scalarized utilities was first studied in the model-based setting by [3, 2], which consider the average-reward setting and establish regret guarantees using explicit model estimation. Further, [4] extends the results with constraints. While these approaches provide strong theoretical guarantees, they rely on estimating the underlying MDP model, which can be computationally expensive in large state-action spaces. A related formulation of scalarized multi-objective RL has been studied in [24]. However, this paper provides an ϵ-approximation guarantee with runtime complexity for a model-based planner, rather than a sample complexity guarantee.

Policy gradient methods for concave utilities. To address the scalability limitations of model-based methods, [7] proposed a model-free policy-gradient algorithm for concave multi-objective reinforcement learning and established an $\mathcal{O}(\epsilon^{-4})$ upper bound on the number of samples needed to compute an ϵ-optimal policy. Subsequent work [34] analyzes natural policy-gradient methods for concave utility reinforcement learning, but assumes access to exact gradients, thereby bypassing the sampling challenges that arise in practical RL settings. The gap between the $\mathcal{O}(\epsilon^{-4})$ guarantees in this literature and the optimal $\mathcal{O}(\epsilon^{-2})$ sample complexity known for standard RL motivates the present work.

Existing MLMC-based natural actor-critic methods for average-reward and constrained MDPs control bias arising from finite Markovian trajectories [12, 29]. In concave-scalarized MORL, an unbiased return estimate is itself insufficient: $\mathbb{E}[\widehat{J}] = J$ does not imply $\mathbb{E}[\partial_m f(\widehat{J})] = \partial_m f(J)$. Controlling trajectory truncation therefore leaves the nonlinear scalarization bias unresolved. We apply MLMC directly to the scalarization gradient, telescoping across nested return batches to attain large-batch bias accuracy at logarithmic expected sampling cost. Our NPG analysis exploits this small bias while accommodating the estimator's larger variance, yielding a $\widetilde{\mathcal{O}}(\epsilon^{-2})$ sample-complexity guarantee under Assumptions 1–5. We also establish a complementary result: under Assumption 6, Taylor cancellation improves the empirical estimator's bias bound from $\mathcal{O}(B^{-1/2})$ to $\mathcal{O}(B^{-1})$, allowing vanilla NPG to attain the same rate without MLMC.

Reinforcement learning with general utilities. Our formulation is also closely related to reinforcement learning with general utilities (RLGU) [32], where the objective is to maximize a concave function $g(\lambda^\pi)$ and $\lambda^\pi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ denotes the discounted state-action occupancy measure of policy π. For finite state and action spaces, the RLGU formulation is equivalent to scalarized multi-objective RL. In particular, if we define $h_m(\lambda) = \langle r_m, \lambda \rangle$ for reward functions r_m and set $g(\lambda) = f(h_1(\lambda), \ldots, h_M(\lambda))$, then $h_m(\lambda^\pi) = J_m^\pi$ and the objective reduces to $f(\mathbf{J}^\pi)$. Conversely, by choosing $M = |\mathcal{S}||\mathcal{A}|$ and indicator rewards $r_{(s,a)}(s', a') = \mathbf{1}\{(s', a') = (s, a)\}$, each return $J^\pi_{(s,a)}$ equals the occupancy $\lambda^\pi(s, a)$ (up to the normalization used in defining λ^π), so any RLGU objective $g(\lambda^\pi)$ can be written as $f(\mathbf{J}^\pi)$ for a correspondingly rescaled f. RLGU has been studied extensively in prior works [15, 31–33, 14], most of which focus on the tabular setting. More recent works extend these results to parametrized policies [9, 16, 22] under structural assumptions such as low-rank MDPs, where the occupancy measure admits a low-dimensional representation. However, even in these regimes the best known global sample complexity for policy-gradient methods remains $\mathcal{O}(\epsilon^{-4})$ [9, 16].

B Empirical Evaluations

We evaluate the proposed MLMC-NPG algorithm (Algorithm 2), building directly on the experimental setup of Bai et al. and benchmarking against both algorithms considered therein [7]: Policy Gradient [7] and an adapted REINFORCE-based algorithm for MORL [7, 28]. To turn episodic environments into infinite horizon MDP, once an episode terminates, the environment enters an absorbing state in which all actions yield zero reward and transition back to the same state. All policy-training results are averaged over 20 random seeds. The experiments were run on a small local workstation with an AMD Ryzen 5 3600X processor (6 cores), 32 GB RAM, and an NVIDIA GeForce RTX 3080 Ti GPU with 12 GB VRAM. Details of the two environments are provided below, and the code is available at `https://github.itap.purdue.edu/Clan-labs/MLMC-NPG-ConcaveMORL`.

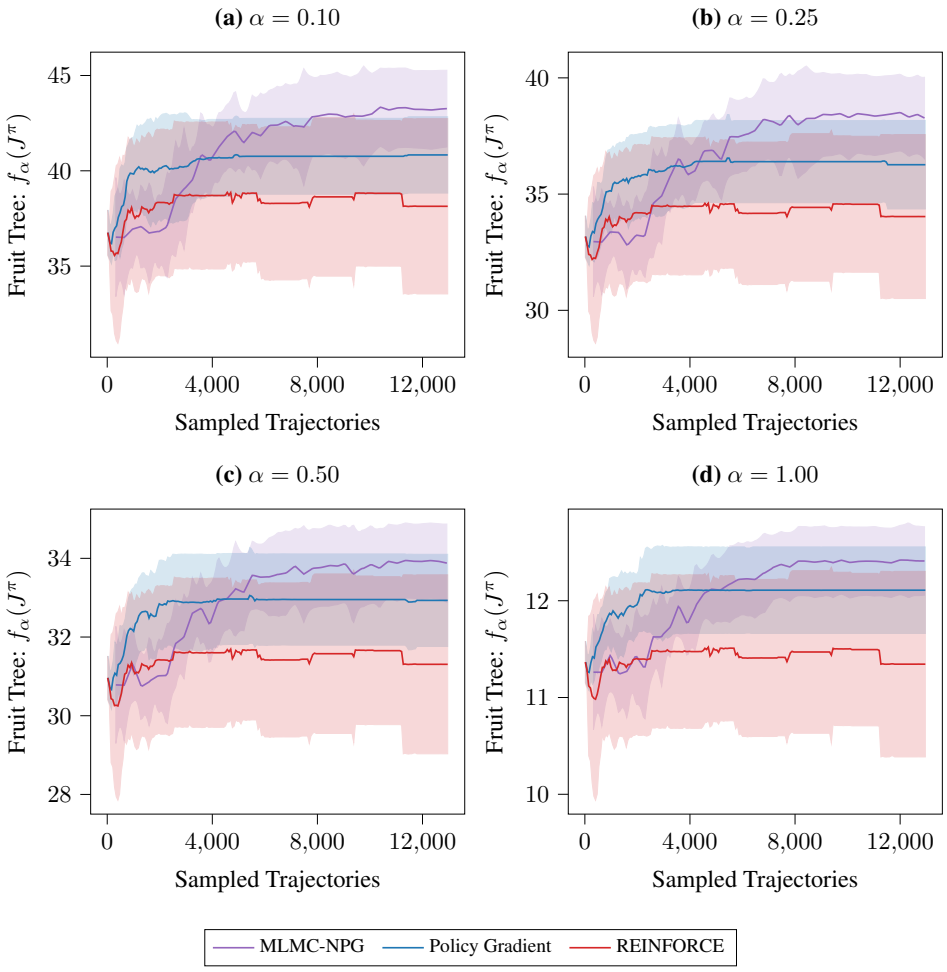

Figure 1: Training curves on Fruit Tree (B=16, MLMC-NPG K=40) performance across four values of α for the α-fairness objective $f_\alpha(J^\pi)$ from Equation (25) with $\alpha \in \{0.10, 0.25, 0.50, 1.00\}$. Iteration counts for Policy Gradient and REINFORCE are scaled to match the total trajectory budget. Shaded regions show ± 1 standard deviation over 20 seeds.

Fruit Tree [30] A full binary tree of depth d=7 with randomly assigned reward $r \in \mathbb{R}^6$ on the leaf nodes denoting different nutrients. We train the agent to maximize different α-fairness objectives (defined concretely in Equation (25)) across M=6 reward components with four values $\alpha \in \{0.10, 0.25, 0.50, 1.00\}$. This environment tests an agent's ability to explore and exploit leaf nodes under criteria that reward different nutrient balances. It captures a common real-world dynamic where the marginal utility of one resource decreases when other resources are scarce. A discount factor γ=0.99 is used, and $H = 7$.

$$f_\alpha(J^\pi) = \begin{cases} \sum_{m=1}^{M} \frac{(J_m^\pi)^{1-\alpha}}{1-\alpha} & \alpha \geq 0,\ \alpha \neq 1 \\ \sum_{m=1}^{M} \log J_m^\pi & \alpha = 1 \end{cases} \tag{25}$$

Breaking Bottles [27] A 1D gridworld environment where the goal is to deliver bottles on the left to the right. There are 3 actions: move left, move right, and pick up a bottle. An episode ends once 2 bottles are successfully delivered or the maximum horizon H=200 is reached. If 2 bottles are held simultaneously, there is a p=0.01 probability of dropping both bottles. If a bottle is dropped,

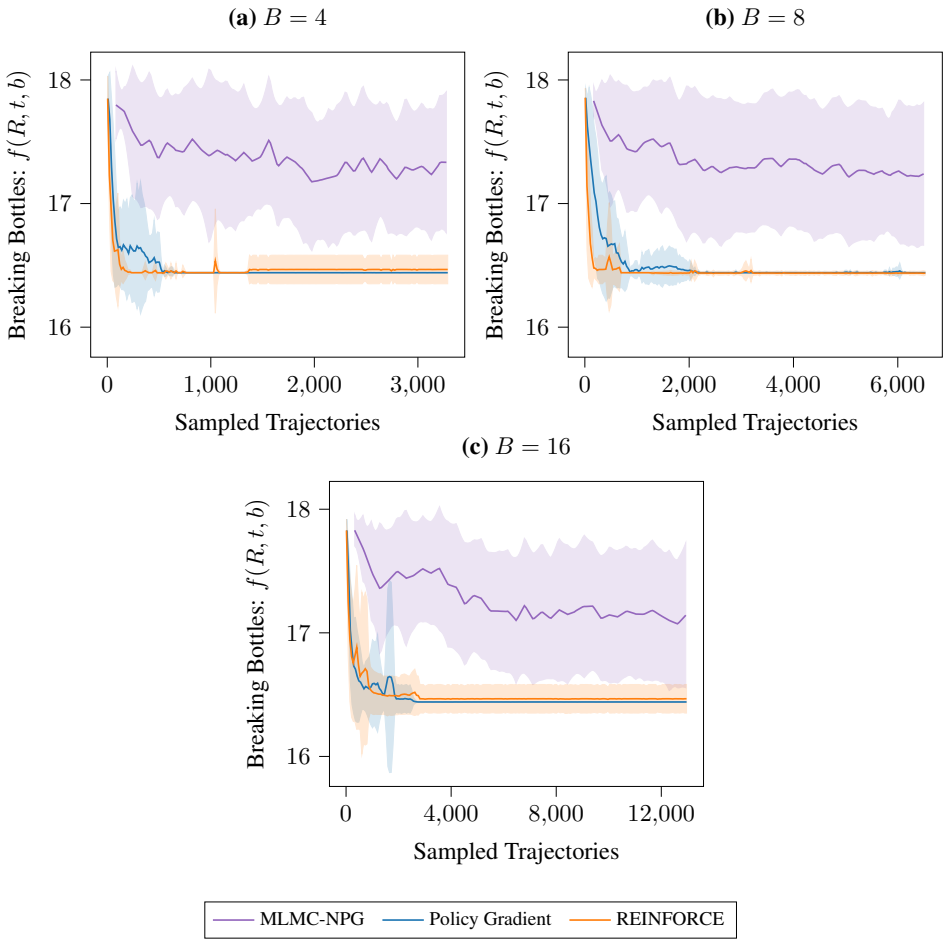

Figure 2: Training curves on Breaking Bottles (H=200, MLMC-NPG K=40) for batch sizes $B \in \{4, 8, 16\}$, optimizing $f(R, t, b)$ as defined in Equation (26). Iteration counts for Policy Gradient and REINFORCE are scaled to match the total trajectory budget. Shaded regions show ± 1 standard deviation over 20 seeds.

then it is broken and cannot be picked back up. World size is 3 blocks. The environment examines an agent's capability to learn a policy to complete a simple task quickly while minimizing risk. A discount factor γ=0.9999 is used. The following reward function minimizing time t and breakages b while maximizing bottles delivered R is utilized:

$$f(R, t, b) = \sqrt{R} + \sqrt{H - t} + \sqrt{H - b} \tag{26}$$

In both environments, the utilities are evaluated on shifted returns, as discussed after Assumption 2: each argument is clipped at zero and shifted by $\delta = 0.5$, i.e., $u_\alpha(\max\{0, J_m^\pi\} + \delta)$ in (25) and $\sqrt{\max\{0, \cdot\} + \delta}$ for each term in (26). This keeps the scalarization gradients bounded on the range of returns encountered.

B.1 Training Parameters

Each algorithm uses the same batch size B and a matched total trajectory budget. Policy Gradient draws two independent batches of size B per iteration and REINFORCE draws one batch of size B per iteration, so their outer iteration counts K are scaled to match the total trajectories consumed by MLMC-NPG. We use $B_{\max} = B$ as each parameter is defined in Algorithm 2. For the policy-training experiments, all policies are parametrized as a neural network with one hidden layer of dimension 32, `tanh` activations, and a softmax output over actions. An Adam optimizer [17] with learning rate 0.05 is used across all algorithms. MLMC-NPG additionally uses an inner-loop step size β=0.05 for

the conjugate-gradient solve approximating the natural gradient, with $N{=}20$ inner steps per outer iteration. The REINFORCE implementation uses the per-step training reward from [7]:

$$r_{\text{train}}(t) = -\sum_{m=1}^{M} \frac{t}{\sum_{\tau=1}^{t} \gamma^{\tau-1}\, r_m(s_\tau, a_\tau)} \tag{27}$$

B.2 Empirical Evaluation Results

Figures 1 and 2 compare mean utility against sampled trajectories across seven configurations. MLMC-NPG achieves higher mean utility than both baselines at the largest reported trajectory budget in every configuration. In Fruit Tree, Policy Gradient initially improves more quickly, while MLMC-NPG achieves higher utility at larger budgets. In Breaking Bottles, MLMC-NPG achieves higher final mean utility for each tested batch size. Appendix B.3 separately examines the scalarization-gradient bias and sampling cost at a fixed policy.

B.3 Fixed-Policy Comparison of Scalarization-Gradient Estimators

We compare the empirical scalarization-gradient estimator used in Algorithm 1 with the MLMC estimator used in Algorithm 2, holding the policy fixed. This isolates the bias and sampling cost of estimating the scalarization coefficients.

We use the depth-seven Fruit Tree environment, with returns scaled to $[0, 1]^6$. Since transitions are deterministic and episodes terminate at one of $2^7 = 128$ leaves, the return vector J^π and the reference scalarization gradient $\partial f(J^\pi)$ can be evaluated exactly by enumerating the terminal paths. The evaluation policy is a frozen tabular softmax policy whose logits are initialized to zero and perturbed by a fixed draw from $0.7\mathcal{N}(0, I)$.

We use the concave utility

$$f(x) = \sum_{m=1}^{M} \left[x_m - \frac{1}{2}\big((x_m - c_m)_+\big)^2 \right], \qquad c_m = J_m^\pi.$$

The constants c_m remain fixed throughout the comparison. This utility has a Lipschitz gradient but does not satisfy Assumption 6. Placing the kink of the scalarization gradient at the evaluation return exposes the plug-in bias.

We sweep $B_1 = B_{\max}$ over powers of two from 1 to 1,024. The empirical estimator uses B_1 trajectories per coefficient estimate, whereas MLMC uses a random number of trajectories. Bias is measured in the infinity norm between the mean estimated scalarization gradient and the exact value $\partial f(J^\pi)$. We estimate these means using 3×10^6 repetitions per configuration. Cost is the mean number of trajectories used for one coefficient estimate.

Table 1 shows similar decreases in estimated bias for the two estimators, with fitted log–log slopes of approximately -0.51 for empirical estimation and -0.50 for MLMC over the tested batch sizes.

The sampling costs differ substantially. At $B_1 = B_{\max} = 1{,}024$, the estimated biases are 2.58×10^{-3} and 2.91×10^{-3}, respectively, while the costs are 1,024 and 9.97 trajectories per estimate. Thus MLMC achieves a comparable estimated bias using approximately 103 times fewer trajectories per coefficient estimate.

For these power-of-two batch sizes, the telescoping construction in Lemma 5 gives the MLMC estimator the same expectation as the corresponding large-batch empirical estimator, while its expected sampling cost grows logarithmically with $B_{\max}$. This experiment illustrates that bias–cost trade-off. The convergence analysis in Theorem 1 additionally accounts for estimator variance and the sampling required by the NPG inner loop.

C Proof of Lemma 1

We begin our analysis by restating the performance difference lemma for concave utilities, which serves as a foundational tool for our derivation.

	Empirical estimator		MLMC estimator	
$B_1 = B_{\max}$	Estimated bias	Mean cost	Estimated bias	Mean cost
1	9.20×10^{-2}	1	9.20×10^{-2}	1.00
2	5.97×10^{-2}	2	5.98×10^{-2}	1.50
4	4.20×10^{-2}	4	4.18×10^{-2}	2.25
8	2.94×10^{-2}	8	2.95×10^{-2}	3.12
16	2.07×10^{-2}	16	2.09×10^{-2}	4.06
32	1.46×10^{-2}	32	1.45×10^{-2}	5.03
64	1.03×10^{-2}	64	1.04×10^{-2}	6.02
128	7.31×10^{-3}	128	7.02×10^{-3}	7.03
256	5.17×10^{-3}	256	5.71×10^{-3}	8.00
512	3.65×10^{-3}	512	3.82×10^{-3}	8.99
1024	2.58×10^{-3}	1024	2.91×10^{-3}	9.97

Table 1: Estimated scalarization-gradient bias and sampling cost at a fixed policy. Cost is measured in trajectories per coefficient estimate; MLMC costs are empirical averages. The reported Monte Carlo standard errors are at most 6.3×10^{-5} for empirical estimation and 2.3×10^{-4} for MLMC.

Lemma 7 (Performance difference lemma for concave utility)**.** *The difference in performance between any two policies π_θ and $\pi_{\theta'}$ is bounded as follows:*

$$(1-\gamma)\left[f(\mathbf{J}^{\pi_\theta}) - f(\mathbf{J}^{\pi_{\theta'}})\right] \leq \sum_{m=1}^{M} \partial_m f(\mathbf{J}^{\pi_{\theta'}}) \mathbb{E}_{s\sim d_\rho^{\pi_\theta}} \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[A_m^{\pi_{\theta'}}(s,a)\right]. \tag{28}$$

This result is adapted from [7] (specifically Lemma 8).

Next, we characterize the progress of the policy update by considering the expected change in the Kullback–Leibler (KL) divergence. Throughout, $\mathbb{E}_*[\cdot]$ denotes $\mathbb{E}_{s\sim d_\rho^{\pi^*}} \mathbb{E}_{a\sim\pi^*(\cdot|s)}[\cdot]$:

$$\begin{aligned}
&\mathbb{E}_{s\sim d_\rho^{\pi^*}}\Big[\mathrm{KL}(\pi^*(\cdot|s)\|\pi_{\theta_k}(\cdot|s)) - \mathrm{KL}(\pi^*(\cdot|s)\|\pi_{\theta_{k+1}}(\cdot|s))\Big] \\
&= \mathbb{E}_*\left[\log\frac{\pi_{\theta_{k+1}}(a|s)}{\pi_{\theta_k}(a|s)}\right] \\
&\overset{(a)}{\geq} \mathbb{E}_*\Big[\nabla_\theta \log\pi_{\theta_k}(a|s)\cdot(\theta_{k+1}-\theta_k)\Big] - \frac{G_2}{2}\|\theta_{k+1}-\theta_k\|^2 \\
&= \alpha\mathbb{E}_*\Big[\nabla_\theta \log\pi_{\theta_k}(a|s)\cdot\omega_k\Big] - \frac{G_2\alpha^2}{2}\|\omega_k\|^2 \\
&= \alpha\mathbb{E}_*\Big[\nabla_\theta \log\pi_{\theta_k}(a|s)\cdot\omega_k^*\Big] + \alpha\mathbb{E}_*\Big[\nabla_\theta \log\pi_{\theta_k}(a|s)\cdot(\omega_k-\omega_k^*)\Big] \\
&\qquad\qquad - \frac{G_2\alpha^2}{2}\|\omega_k\|^2 \\
&= \alpha\big[f(\mathbf{J}^*) - f(\mathbf{J}^{\pi_{\theta_k}})\big] + \alpha\mathbb{E}_*\Big[\nabla_\theta \log\pi_{\theta_k}(a|s)\cdot\omega_k^*\Big] - \alpha\big[f(\mathbf{J}^*) - f(\mathbf{J}^{\pi_{\theta_k}})\big] \\
&\qquad\qquad + \alpha\mathbb{E}_*\Big[\nabla_\theta \log\pi_{\theta_k}(a|s)\cdot(\omega_k-\omega_k^*)\Big] - \frac{G_2\alpha^2}{2}\|\omega_k\|^2 \\
&\overset{(b)}{\geq} \alpha\big[f(\mathbf{J}^*) - f(\mathbf{J}^{\pi_{\theta_k}})\big] + \alpha\mathbb{E}_*\Big[\nabla_\theta \log\pi_{\theta_k}(a|s)\cdot(\omega_k-\omega_k^*)\Big] - \frac{G_2\alpha^2}{2}\|\omega_k\|^2 \\
&\qquad\qquad + \frac{\alpha}{1-\gamma}\mathbb{E}_*\left[(1-\gamma)\nabla_\theta \log\pi_{\theta_k}(a|s)\cdot\omega_k^* - \sum_{m=1}^{M}\partial_m f(\mathbf{J}^{\pi_{\theta_k}}) A_m^{\pi_{\theta_k}}(s,a)\right] \\
&\overset{(c)}{\geq} \alpha\big[f(\mathbf{J}^{\pi^*}) - f(\mathbf{J}^{\pi_{\theta_k}})\big] + \alpha\mathbb{E}_*\Big[\nabla_\theta \log\pi_{\theta_k}(a|s)\cdot(\omega_k-\omega_k^*)\Big] - \frac{G_2\alpha^2}{2}\|\omega_k\|^2 \\
&\qquad\qquad - \frac{\alpha}{1-\gamma}\sqrt{\mathbb{E}_*\left[\left((1-\gamma)\nabla_\theta \log\pi_{\theta_k}(a|s)\cdot\omega_k^* - \sum_{m=1}^{M}\partial_m f(\mathbf{J}^{\pi_{\theta_k}}) A_m^{\pi_{\theta_k}}(s,a)\right)^2\right]}
\end{aligned} \tag{29}$$

where step (a) holds by Assumption 3 and step (b) holds by Lemma 7. Step (c) uses the convexity of the function $f(x) = x^2$. Rearranging items, we have

$$f(\mathbf{J}^*) - \mathbb{E}\, f(\mathbf{J}^{\pi_{\theta_k}}) \leq \frac{\sqrt{2\varepsilon_{\text{bias}}}}{1-\gamma} + G_1 \mathbb{E}\, \|\mathbb{E}[\omega_k|\theta_k] - \omega_k^*\| + \frac{G_2\alpha}{2}\, \mathbb{E}\, \|\omega_k\|^2 + \frac{1}{\alpha}\, \mathbb{E}\, \mathbb{E}_{s\sim d_\rho^{\pi^*}}\Big[\, \mathrm{KL}(\pi^*(\cdot|s)\|\pi_{\theta_k}(\cdot|s)) - \mathrm{KL}(\pi^*(\cdot|s)\|\pi_{\theta_{k+1}}(\cdot|s))\Big]. \quad (30)$$

Summing from $k = 0$ to $K - 1$ and dividing by K concludes the proof of Lemma 1.

D Proof of Lemma 8

Next, we bound the average squared gradient norm. Here, we use that $f(\mathbf{J}^{\pi_\theta})$ is L_J-smooth in θ (see Lemma 9).

Lemma 8 (Stationary Convergence)**.** *Suppose the parameter update satisfies $\theta_{k+1} = \theta_k + \alpha\omega_k$. Under Assumptions 2–4, we have*

$$\left(\frac{\alpha}{2G_1^2} - \frac{L_J\alpha^2}{\mu^2}\right)\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|\nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})\|^2 \quad \leq \frac{\mathbb{E}[f(\mathbf{J}^{\pi_{\theta_K}}) - f(\mathbf{J}^{\pi_{\theta_0}})]}{K} + \frac{\alpha G_1^2}{2K}\sum_{k=0}^{K-1}\mathbb{E}\,\|\mathbb{E}[\omega_k|\theta_k] - \omega_k^*\|^2 + \frac{L_J\alpha^2}{K}\sum_{k=0}^{K-1}\mathbb{E}\|\omega_k - \omega_k^*\|^2. \quad (31)$$

where

$$L_J = MC\left(\frac{G_2}{(1-\gamma)^2} + \frac{(1+\gamma)G_1^2}{(1-\gamma)^3}\right) + \frac{M^{3/2}L_f G_1^2}{(1-\gamma)^4}. \quad (32)$$

We next bound objective smoothness and horizon truncation.

Lemma 9. *Suppose Assumptions 2 and 3 hold. Both the infinite-horizon objective $f(\mathbf{J}^{\pi_\theta})$ and the truncated objective $f(\mathbf{J}_H^{\pi_\theta})$ are L_J-smooth in θ, where*

$$L_J = MC\left(\frac{G_2}{(1-\gamma)^2} + \frac{(1+\gamma)G_1^2}{(1-\gamma)^3}\right) + \frac{M^{3/2}L_f G_1^2}{(1-\gamma)^4}. \quad (33)$$

Furthermore, the gradients of the original and truncated objectives satisfy

$$\|\nabla_\theta f(\mathbf{J}^{\pi_\theta}) - \nabla_\theta f(\mathbf{J}_H^{\pi_\theta})\| \leq \frac{MG_1\gamma^H}{(1-\gamma)^2}\left[\sqrt{M}L_f\frac{1-\gamma^H - H\gamma^H(1-\gamma)}{1-\gamma} + C(1+H(1-\gamma))\right]. \quad (34)$$

For each return coordinate,

$$\|\nabla_\theta \mathbf{J}_m^{\pi_\theta}\| \leq \frac{G_1}{(1-\gamma)^2}, \qquad \mathrm{Lip}(\nabla_\theta \mathbf{J}_m^{\pi_\theta}) \leq \frac{G_2}{(1-\gamma)^2} + \frac{(1+\gamma)G_1^2}{(1-\gamma)^3}.$$

These bounds follow by differentiating trajectory likelihoods and summing $(t+1)\gamma^t$ and $(t+1)^2\gamma^t$. For Lipschitz scores, the derivative bounds hold almost everywhere and yield the same Lipschitz estimates by integration.

Expanding $\nabla_\theta f(\mathbf{J}^{\pi_\theta}) = \sum_m \partial_m f(\mathbf{J}^{\pi_\theta})\nabla_\theta \mathbf{J}_m^{\pi_\theta}$ then gives the stated L_J: variation of the return gradients gives the first term, and variation of the scalarization derivatives gives the final term. The same bounds hold for truncated returns.

Using the L_J-smoothness of $f(\mathbf{J}^{\pi_\theta})$ with respect to θ (Lemma 9), we obtain

$$\begin{aligned}
f(\mathbf{J}^{\pi_{\theta_{k+1}}}) &\geq f(\mathbf{J}^{\pi_{\theta_k}}) + \langle \nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}}),\, \theta_{k+1} - \theta_k \rangle - \frac{L_J}{2}\|\theta_{k+1} - \theta_k\|^2 \\
&= f(\mathbf{J}^{\pi_{\theta_k}}) + \alpha \langle \nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}}),\, \omega_k \rangle - \frac{L_J \alpha^2}{2}\|\omega_k\|^2 \\
&= f(\mathbf{J}^{\pi_{\theta_k}}) + \alpha \langle \nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}}),\, \omega_k - \omega_k^* + \omega_k^* \rangle - \frac{L_J \alpha^2}{2}\|\omega_k - \omega_k^* + \omega_k^*\|^2 \\
&\geq f(\mathbf{J}^{\pi_{\theta_k}}) + \alpha \langle \nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}}),\, \omega_k^* \rangle + \alpha \langle \nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}}),\, \omega_k - \omega_k^* \rangle \\
&\qquad - L_J \alpha^2 \left(\|\omega_k - \omega_k^*\|^2 + \|\omega_k^*\|^2 \right) \\
&\overset{(a)}{\geq} f(\mathbf{J}^{\pi_{\theta_k}}) + \frac{\alpha}{G_1^2}\|\nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})\|^2 + \alpha \langle \nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}}),\, \omega_k - \omega_k^* \rangle \\
&\qquad - L_J \alpha^2 \|\omega_k - \omega_k^*\|^2 - L_J \alpha^2 \|\omega_k^*\|^2 \\
&\overset{(b)}{\geq} f(\mathbf{J}^{\pi_{\theta_k}}) + \left(\frac{\alpha}{G_1^2} - \frac{\alpha^2 L_J}{\mu^2} \right) \|\nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})\|^2 + \alpha \langle \nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}}),\, \omega_k - \omega_k^* \rangle \\
&\qquad - L_J \alpha^2 \|\omega_k - \omega_k^*\|^2. \qquad (35)
\end{aligned}$$

Taking conditional expectation given θ_k, we have

$$\begin{aligned}
\mathbb{E}[f(\mathbf{J}^{\pi_{\theta_{k+1}}})|\theta_k] &\geq f(\mathbf{J}^{\pi_{\theta_k}}) + \left(\frac{\alpha}{G_1^2} - \frac{\alpha^2 L_J}{\mu^2} \right) \|\nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})\|^2 \\
&\qquad + \alpha \langle \nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}}),\, \mathbb{E}[\omega_k|\theta_k] - \omega_k^* \rangle - L_J \alpha^2 \,\mathbb{E}\big[\|\omega_k - \omega_k^*\|^2 | \theta_k\big] \\
&\geq f(\mathbf{J}^{\pi_{\theta_k}}) + \left(\frac{\alpha}{G_1^2} - \frac{\alpha^2 L_J}{\mu^2} \right) \|\nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})\|^2 \\
&\qquad - \alpha \left(c\|\nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})\|^2 + \frac{1}{4c}\|\,\mathbb{E}[\omega_k|\theta_k] - \omega_k^*\|^2 \right) \\
&\qquad - L_J \alpha^2 \,\mathbb{E}\big[\|\omega_k - \omega_k^*\|^2 | \theta_k\big] \\
&= f(\mathbf{J}^{\pi_{\theta_k}}) + \left(\frac{\alpha}{G_1^2} - \frac{\alpha^2 L_J}{\mu^2} - c\alpha \right) \|\nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})\|^2 \\
&\qquad - \frac{\alpha}{4c}\|\,\mathbb{E}[\omega_k|\theta_k] - \omega_k^*\|^2 - L_J \alpha^2 \,\mathbb{E}\big[\|\omega_k - \omega_k^*\|^2 | \theta_k\big]. \qquad (36)
\end{aligned}$$

Here, step (a) follows from Assumption 3, and step (b) follows from Assumption 4. The second inequality after taking conditional expectation follows from Young's inequality. We obtain the following by choosing $c = 1/(2G_1^2)$ and taking expectation over θ_k:

$$\begin{aligned}
\mathbb{E}[f(\mathbf{J}^{\pi_{\theta_{k+1}}})] &\geq \mathbb{E}[f(\mathbf{J}^{\pi_{\theta_k}})] + \left(\frac{\alpha}{2G_1^2} - \frac{\alpha^2 L_J}{\mu^2} \right) \mathbb{E}\,\|\nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})\|^2 \\
&\qquad - \frac{\alpha G_1^2}{2}\,\mathbb{E}\,\|\,\mathbb{E}[\omega_k|\theta_k] - \omega_k^*\|^2 - L_J \alpha^2 \,\mathbb{E}\big[\|\omega_k - \omega_k^*\|^2\big]. \qquad (37)
\end{aligned}$$

Finally, rearranging and taking average over $k = 0, \ldots, K-1$ yields Lemma 8.

E Proof of Lemma 2

Fix the current policy throughout this proof. Expectations average over return estimation and inner-loop sampling.

For the Fisher estimator,

$$0 \preceq \widehat{F}_n \preceq \Lambda_F I, \qquad \Lambda_F = G_1^2.$$

Each Fisher batch is independent of the preceding inner-loop history. Its expectation does not depend on n and satisfies

$$\|\mathbb{E}[\widehat{F}_n] - F\| \leq \delta_F = G_1^2 \gamma^H.$$

Choose H so that $\delta_F \le \mu/2$. Then $\mathbb{E}[\widehat{F}_n] \succeq (\mu/2)I$.

The one-trajectory Fisher estimator is positive semidefinite and has trace at most G_1^2. Applying the variance identity in Frobenius norm, which upper-bounds operator-norm MSE, therefore gives

$$\mathbb{E}\|\widehat{F}_n - F\|^2 \le G_1^4 \left(\frac{1}{B} + \gamma^{2H}\right) =: \sigma_F^2.$$

Here B denotes the inner-loop batch size.

Let $\omega^* = F^{-1}\nabla_\theta f(J^{\pi_\theta})$ and $R_0 = \|\omega_0 - \omega^*\|$. Also,

$$\|\nabla_\theta f(J^{\pi_\theta})\| \le \frac{CMG_1}{(1-\gamma)^2} =: \Lambda_g.$$

Use $\beta = \mu/(4\Lambda_F^2)$.

Fresh Fisher batches give the mean recursion

$$\begin{aligned}\mathbb{E}[\omega_{n+1}] - \omega^* = (I - \beta\mathbb{E}[\widehat{F}_n])(\mathbb{E}[\omega_n] - \omega^*) \\ + \beta\big(\mathbb{E}[\widehat{g}_n] - \mathbb{E}[\widehat{F}_n]\omega^*\big).\end{aligned}$$

The forcing term has norm at most $\bar{\delta}_g + \delta_F\Lambda_g/\mu$. Unrolling consequently gives

$$\|\mathbb{E}[\omega_N] - \omega^*\| \le e^{-N\mu^2/(8\Lambda_F^2)}R_0 + \frac{2\bar{\delta}_g}{\mu} + \frac{2\delta_F\Lambda_g}{\mu^2}.$$

For the mean-square error, positivity and boundedness of $\widehat{F}_n$ imply

$$\begin{aligned}\mathbb{E}\|(I - \beta\widehat{F}_n)(\omega_n - \omega^*)\|^2 \\ \le (1 - \beta\mu/2)\mathbb{E}\|\omega_n - \omega^*\|^2.\end{aligned}$$

Expanding the update and applying Young's inequality with parameter $\beta\mu/4$ therefore yields

$$\begin{aligned}\mathbb{E}\|\omega_{n+1} - \omega^*\|^2 \le (1 - \beta\mu/4)\mathbb{E}\|\omega_n - \omega^*\|^2 \\ + \frac{5\beta}{\mu}\mathbb{E}\|\widehat{g}_n - \widehat{F}_n\omega^*\|^2.\end{aligned}$$

Moreover,

$$\mathbb{E}\|\widehat{g}_n - \widehat{F}_n\omega^*\|^2 \le 2\sigma_g^2 + \frac{2\sigma_F^2\Lambda_g^2}{\mu^2}.$$

Unrolling gives

$$\mathbb{E}\|\omega_N - \omega^*\|^2 \le e^{-N\mu^2/(16\Lambda_F^2)}R_0^2 + \frac{40\sigma_g^2}{\mu^2} + \frac{40\sigma_F^2\Lambda_g^2}{\mu^4}.$$

No independence between the Fisher and gradient estimates within the same batch is required. These bounds imply Lemma 2 with $c = \mu^2/(16G_1^4)$.

F Proof of Lemma 3

Recall that the gradient estimator is defined as:

$$g(\tau_i^H, \widehat{\mathbf{J}}_H^{\pi_\theta}|\theta) := \sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(a_t^i|s_t^i) \left(\sum_{m=1}^{M} \partial_m f(\widehat{\mathbf{J}}_H^{\pi_\theta}) \sum_{h=t}^{H-1} \gamma^h r_m(s_h^i, a_h^i)\right). \tag{38}$$

The expectation of the estimator with respect to the truncated horizon objective is given by:

$$\mathbb{E}[g(\tau_i^H, \mathbf{J}_H^{\pi_\theta}|\theta)] = \mathbb{E}\left[\sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(a_t^i|s_t^i) \left(\sum_{m=1}^{M} \partial_m f(\mathbf{J}_H^{\pi_\theta}) \sum_{h=t}^{H-1} \gamma^h r_m(s_h^i, a_h^i)\right)\right]. \tag{39}$$

Invoking Lemma 9, we observe that the truncation error vanishes exponentially in H:

$$\|\mathbb{E}[g(\tau_i^H, \mathbf{J}_H^{\pi_\theta}|\theta)] - \nabla_\theta f(\mathbf{J}^{\pi_\theta})\| \leq \mathcal{O}((1+H)\gamma^H). \tag{40}$$

To bound the total bias, we decompose the error into terms reflecting the estimation error for the truncated objective and the horizon truncation error:

$$\begin{aligned}\mathbb{E}\left[g(\tau_i^H, \widehat{\mathbf{J}}_H^{\pi_\theta}|\theta)\right] - \nabla_\theta f(\mathbf{J}^{\pi_\theta}) = \mathbb{E}\left[g(\tau_i^H, \widehat{\mathbf{J}}_H^{\pi_\theta}|\theta) - g(\tau_i^H, \mathbf{J}_H^{\pi_\theta}|\theta)\right] \\ + \mathbb{E}\left[g(\tau_i^H, \mathbf{J}_H^{\pi_\theta}|\theta)\right] - \nabla_\theta f(\mathbf{J}^{\pi_\theta}).\end{aligned} \tag{41}$$

For the first term on the right-hand side, we have:

$$\begin{aligned}&\left\|\mathbb{E}[g(\tau_i^H, \mathbf{J}_H^{\pi_\theta}|\theta)] - \mathbb{E}[g(\tau_i^H, \widehat{\mathbf{J}}_H^{\pi_\theta}|\theta)]\right\| \\ &= \left\|\mathbb{E}\left[\sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(a_t^i|s_t^i) \sum_{m=1}^{M} \left(\partial_m f(\mathbf{J}_H^{\pi_\theta}) - \partial_m f(\widehat{\mathbf{J}}_H^{\pi_\theta})\right) \hat{Q}_{m,t}\right]\right\|,\end{aligned} \tag{42}$$

where $\hat{Q}_{m,t} = \sum_{h=t}^{H-1} \gamma^h r_m(s_h^i, a_h^i)$. If the trajectories used for $\widehat{\mathbf{J}}_H^{\pi_\theta}$ are independent of τ_i^H, the expectation factors, allowing us to bound the terms separately.

$$\begin{aligned}&\|\,\mathbb{E}[g(\tau_i^H, \mathbf{J}_H^{\pi_\theta}|\theta)] - \mathbb{E}[g(\tau_i^H, \widehat{\mathbf{J}}_H^{\pi_\theta}|\theta)]\| \\ &= \left\|\mathbb{E}\left[\sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(a_t^i|s_t^i) \left(\sum_{m=1}^{M} \left(\partial_m f(\mathbf{J}_H^{\pi_\theta}) - \partial_m f\left(\widehat{\mathbf{J}}_H^{\pi_\theta}\right)\right) \sum_{h=t}^{H-1} \gamma^h r_m(s_h^i, a_h^i)\right)\right]\right\| \\ &= \left\|\sum_{t=0}^{H-1} \sum_{m=1}^{M} \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a_t^i|s_t^i) \left(\sum_{h=t}^{H-1} \gamma^h r_m(s_h^i, a_h^i)\right) \left(\partial_m f(\mathbf{J}_H^{\pi_\theta}) - \partial_m f\left(\widehat{\mathbf{J}}_H^{\pi_\theta}\right)\right)\right]\right\| \\ &= \left\|\sum_{t=0}^{H-1} \sum_{m=1}^{M} \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a_t^i|s_t^i) \left(\sum_{h=t}^{H-1} \gamma^h r_m(s_h^i, a_h^i)\right)\right] \mathbb{E}\left[\partial_m f(\mathbf{J}_H^{\pi_\theta}) - \partial_m f\left(\widehat{\mathbf{J}}_H^{\pi_\theta}\right)\right]\right\| \\ &\leq \sum_{t=0}^{H-1} \sum_{m=1}^{M} \left\|\mathbb{E}\left[\nabla_\theta \log \pi_\theta(a_t^i|s_t^i) \left(\sum_{h=t}^{H-1} \gamma^h r_m(s_h^i, a_h^i)\right)\right]\right\| \cdot \left|\partial_m f(\mathbf{J}_H^{\pi_\theta}) - \mathbb{E}\left[\partial_m f\left(\widehat{\mathbf{J}}_H^{\pi_\theta}\right)\right]\right|\end{aligned} \tag{43}$$

Using boundedness of the score function and rewards, we obtain the following bound:

$$\begin{aligned}\left\|\mathbb{E}\left[\nabla_\theta \log \pi_\theta(a_t^i|s_t^i) \left(\sum_{h=t}^{H-1} \gamma^h r_m(s_h^i, a_h^i)\right)\right]\right\| &\leq \|\nabla_\theta \log \pi_\theta(a_t^i|s_t^i)\| \left|\sum_{h=t}^{H-1} \gamma^h r_m(s_h^i, a_h^i)\right| \\ &\leq G_1 \sum_{h=t}^{\infty} \gamma^h = \frac{G_1 \gamma^t}{1-\gamma}\end{aligned} \tag{44}$$

Thus, we obtain

$$\begin{aligned}\|\,\mathbb{E}[g(\tau_i^H, \mathbf{J}_H^{\pi_\theta}|\theta)] - \mathbb{E}[g(\tau_i^H, \widehat{\mathbf{J}}_H^{\pi_\theta}|\theta)]\| &\leq \sum_{t=0}^{H-1} \frac{G_1 \gamma^t}{1-\gamma} \sum_{m=1}^{M} \left|\partial_m f(\mathbf{J}_H^{\pi_\theta}) - \mathbb{E}\left[\partial_m f\left(\widehat{\mathbf{J}}_H^{\pi_\theta}\right)\right]\right| \\ &\leq \frac{G_1}{(1-\gamma)^2} \sum_{m=1}^{M} \left|\partial_m f(\mathbf{J}_H^{\pi_\theta}) - \mathbb{E}\left[\partial_m f\left(\widehat{\mathbf{J}}_H^{\pi_\theta}\right)\right]\right|\end{aligned} \tag{45}$$

Similarly, we can bound the second order expected difference as follows

$$\begin{aligned}&\mathbb{E}\,\|g(\tau_i^H, \mathbf{J}_H^{\pi_\theta}|\theta) - g(\tau_i^H, \widehat{\mathbf{J}}_H^{\pi_\theta}|\theta)\|^2 \\ &= \mathbb{E}\left\|\sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(a_t^i|s_t^i) \left(\sum_{m=1}^{M} \left(\partial_m f(\mathbf{J}_H^{\pi_\theta}) - \partial_m f\left(\widehat{\mathbf{J}}_H^{\pi_\theta}\right)\right) \sum_{h=t}^{H-1} \gamma^h r_m(s_h^i, a_h^i)\right)\right\|^2 \\ &= \mathbb{E}\left\|\sum_{t=0}^{H-1} \sum_{m=1}^{M} \nabla_\theta \log \pi_\theta(a_t^i|s_t^i) \left(\sum_{h=t}^{H-1} \gamma^h r_m(s_h^i, a_h^i)\right) \left(\partial_m f(\mathbf{J}_H^{\pi_\theta}) - \partial_m f\left(\widehat{\mathbf{J}}_H^{\pi_\theta}\right)\right)\right\|^2\end{aligned} \tag{46}$$

Note that

$$\begin{aligned}
&\left\|\sum_{t=0}^{H-1}\sum_{m=1}^{M}\nabla_\theta \log \pi_\theta(a_t^i|s_t^i)\left(\sum_{h=t}^{H-1}\gamma^h r_m(s_h^i,a_h^i)\right)\left(\partial_m f(\mathbf{J}_H^{\pi_\theta})-\partial_m f\left(\widehat{\mathbf{J}}_H^{\pi_\theta}\right)\right)\right\| \\
&\le \sum_{t=0}^{H-1}\sum_{m=1}^{M}\|\nabla_\theta \log \pi_\theta(a_t^i|s_t^i)\|\left|\sum_{h=t}^{H-1}\gamma^h r_m(s_h^i,a_h^i)\right|\left|\partial_m f(\mathbf{J}_H^{\pi_\theta})-\partial_m f\left(\widehat{\mathbf{J}}_H^{\pi_\theta}\right)\right| \\
&\le \sum_{t=0}^{H-1}\sum_{m=1}^{M}\frac{G_1\gamma^t}{1-\gamma}\left|\partial_m f(\mathbf{J}_H^{\pi_\theta})-\partial_m f\left(\widehat{\mathbf{J}}_H^{\pi_\theta}\right)\right| \\
&\le \frac{G_1}{(1-\gamma)^2}\sum_{m=1}^{M}\left|\partial_m f(\mathbf{J}_H^{\pi_\theta})-\partial_m f\left(\widehat{\mathbf{J}}_H^{\pi_\theta}\right)\right| \qquad (47)
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathbb{E}\,\|g(\tau_i^H,\mathbf{J}_H^{\pi_\theta}|\theta)-g(\tau_i^H,\widehat{\mathbf{J}}_H^{\pi_\theta}|\theta)\|^2 &\le \mathbb{E}\left(\frac{G_1}{(1-\gamma)^2}\sum_{m=1}^{M}\left|\partial_m f(\mathbf{J}_H^{\pi_\theta})-\partial_m f\left(\widehat{\mathbf{J}}_H^{\pi_\theta}\right)\right|\right)^2 \\
&\le \frac{G_1^2 M}{(1-\gamma)^4}\sum_{m=1}^{M}\mathbb{E}\left|\partial_m f(\mathbf{J}_H^{\pi_\theta})-\partial_m f\left(\widehat{\mathbf{J}}_H^{\pi_\theta}\right)\right|^2 \qquad (48)
\end{aligned}$$

G Proof of Lemma 4

For a single trajectory τ_j^H, let $X_m^j=\sum_{t=0}^{H-1}\gamma^t r_m(s_t^j,a_t^j)$. Since $r_m\in[0,1]$, we have $X_m^j\in[0,\frac{1}{1-\gamma}]$ and $\mathbb{E}[X_m^j]=[\mathbf{J}_H^{\pi_\theta}]_m$, so $\mathrm{Var}(X_m^j)\le\mathbb{E}[(X_m^j)^2]\le(1-\gamma)^{-2}$. Since the B trajectories are i.i.d.,

$$\mathbb{E}\left\|\widehat{\mathbf{J}}_{H,B}^{\pi_\theta}-\mathbf{J}_H^{\pi_\theta}\right\|^2=\sum_{m=1}^{M}\mathrm{Var}\left(\frac{1}{B}\sum_{j=1}^{B}X_m^j\right)\le\frac{M}{(1-\gamma)^2 B}.$$

H Proof of Lemma 5

Let $q_{\max}=\lfloor\log_2 B_{\max}\rfloor$, so that $2^Q\le B_{\max}$ if and only if $Q\le q_{\max}$, and write $w_m=\partial_m f(\widehat{\mathbf{J}}_{H,\mathrm{MLMC}}^{\pi_\theta})$ for the MLMC weight defined in (14).

Part (i). The number of sampled trajectories is $B_k=2^{\bar{Q}_k}$, which equals 2^{Q_k} if $Q_k\le q_{\max}$ and 1 otherwise. Hence

$$\mathbb{E}[B_k]=\sum_{q=1}^{q_{\max}}2^{-q}\,2^q+\Pr(Q_k>q_{\max})=q_{\max}+2^{-q_{\max}}=\mathcal{O}(\log B_{\max}).$$

Part (ii). For every b, $\widehat{\mathbf{J}}_{H,b}^{\pi_\theta}$ is an average of b i.i.d. truncated returns, so its distribution depends only on b. In particular, this holds for the nested coarse batch (the first half of the fine batch) and for the base batch (its first trajectory). Taking expectations in (14) and telescoping gives

$$\begin{aligned}
\mathbb{E}[w_m]&=\mathbb{E}\left[\partial_m f(\widehat{\mathbf{J}}_{H,1}^{\pi_\theta})\right]+\sum_{q=1}^{q_{\max}}2^{-q}\,2^q\Big(\mathbb{E}\left[\partial_m f(\widehat{\mathbf{J}}_{H,2^q}^{\pi_\theta})\right]-\mathbb{E}\left[\partial_m f(\widehat{\mathbf{J}}_{H,2^{q-1}}^{\pi_\theta})\right]\Big) \\
&=\mathbb{E}\left[\partial_m f(\widehat{\mathbf{J}}_{H,2^{q_{\max}}}^{\pi_\theta})\right].
\end{aligned}$$

By Assumption 2, Jensen's inequality, Lemma 4, and $2^{q_{\max}}>B_{\max}/2$,

$$\left|\partial_m f(\mathbf{J}_H^{\pi_\theta})-\mathbb{E}[w_m]\right|\le L_f\;\mathbb{E}\left\|\widehat{\mathbf{J}}_{H,2^{q_{\max}}}^{\pi_\theta}-\mathbf{J}_H^{\pi_\theta}\right\|\le\frac{L_f\sqrt{M}}{(1-\gamma)\sqrt{2^{q_{\max}}}}\le\frac{L_f\sqrt{2M}}{(1-\gamma)\sqrt{B_{\max}}}.$$

The MLMC batch is independent of the gradient trajectory τ_i^H, so the argument in the proof of Lemma 3(i) applies with $\partial_m f(\widehat{\mathbf{J}}_H^{\pi_\theta})$ replaced by w_m. Summing the bound above over $m\in[M]$ and adding the truncation error proves part (ii).

Part (iii). We compute a second-order moment bound:

$$\mathbb{E}\,|w_m|^2 = \mathbb{E}\left|\partial_m f\left(\widehat{\mathbf{J}}^{\pi_\theta}_{H,1}\right) + \mathbf{1}_{\{2^Q \le B_{\max}\}}\, 2^Q\left(\partial_m f\left(\widehat{\mathbf{J}}^{\pi_\theta}_{H,2^Q}\right) - \partial_m f\left(\widehat{\mathbf{J}}^{\pi_\theta}_{H,2^{Q-1}}\right)\right)\right|^2$$
$$\le 2\,\mathbb{E}\left|\partial_m f\left(\widehat{\mathbf{J}}^{\pi_\theta}_{H,1}\right)\right|^2 + 2\,\mathbb{E}\left|\mathbf{1}_{\{2^Q \le B_{\max}\}}\, 2^Q\left(\partial_m f\left(\widehat{\mathbf{J}}^{\pi_\theta}_{H,2^Q}\right) - \partial_m f\left(\widehat{\mathbf{J}}^{\pi_\theta}_{H,2^{Q-1}}\right)\right)\right|^2$$

Note that

$$\begin{aligned}
&\mathbb{E}\left|\mathbf{1}_{\{2^Q \le B_{\max}\}}\, 2^Q\left(\partial_m f\left(\widehat{\mathbf{J}}^{\pi_\theta}_{H,2^Q}\right) - \partial_m f\left(\widehat{\mathbf{J}}^{\pi_\theta}_{H,2^{Q-1}}\right)\right)\right|^2 \\
&= \sum_{q=1}^{\lfloor \log_2 B_{\max}\rfloor} \Pr(Q=q) 4^q\, \mathbb{E}\left(\partial_m f\left(\widehat{\mathbf{J}}^{\pi_\theta}_{H,2^q}\right) - \partial_m f\left(\widehat{\mathbf{J}}^{\pi_\theta}_{H,2^{q-1}}\right)\right)^2 \\
&\le \sum_{q=1}^{\lfloor \log_2 B_{\max}\rfloor} 2^q L_f^2\, \mathbb{E}\left\|\widehat{\mathbf{J}}^{\pi_\theta}_{H,2^q} - \widehat{\mathbf{J}}^{\pi_\theta}_{H,2^{q-1}}\right\|^2 \\
&\le \sum_{q=1}^{\lfloor \log_2 B_{\max}\rfloor} 2^{q+1} L_f^2 \left(\mathbb{E}\left\|\widehat{\mathbf{J}}^{\pi_\theta}_{H,2^q} - \mathbf{J}^{\pi_\theta}_H\right\|^2 + \mathbb{E}\left\|\mathbf{J}^{\pi_\theta}_H - \widehat{\mathbf{J}}^{\pi_\theta}_{H,2^{q-1}}\right\|^2\right) \\
&\le \sum_{q=1}^{\lfloor \log_2 B_{\max}\rfloor} 2^{q+1} L_f^2 \left(\frac{M}{(1-\gamma)^2 2^q} + \frac{M}{(1-\gamma)^2 2^{q-1}}\right) \le \frac{6ML_f^2 \lfloor \log_2 B_{\max}\rfloor}{(1-\gamma)^2}
\end{aligned} \tag{49}$$

Including the base coefficient contribution and using the coefficient second moments above gives

$$\mathbb{E}\|g(\tau_i^H, \widehat{\mathbf{J}}^{\pi_\theta}_{H,\mathrm{MLMC}} \mid \theta)\|^2 \le \frac{2C^2 G_1^2 M^2}{(1-\gamma)^4} + \frac{12 L_f^2 G_1^2 M^3 \lfloor \log_2 B_{\max}\rfloor}{(1-\gamma)^6}.$$

Combining this with $\|\nabla_\theta f(\mathbf{J}^{\pi_\theta})\| \le CMG_1/(1-\gamma)^2$ and $\|a-b\|^2 \le 2\|a\|^2 + 2\|b\|^2$ yields

$$\mathbb{E}\left\|\nabla_\theta f(\mathbf{J}^{\pi_\theta}) - g(\tau_i^H, \widehat{\mathbf{J}}^{\pi_\theta}_{H,\mathrm{MLMC}}|\theta)\right\|^2 \le \frac{6C^2 G_1^2 M^2}{(1-\gamma)^4} + \frac{24 L_f^2 G_1^2 M^3 \lfloor \log_2 B_{\max}\rfloor}{(1-\gamma)^6}, \tag{50}$$

and concludes the proof of part (iii) of Lemma 5.

I Proof of Lemma 6

Since $\partial_m f$ is $L_{2,f}$-smooth for each $m \in [M]$ by Assumption 6, we have:

$$\left|\partial_m f(\widehat{\mathbf{J}}^{\pi_\theta}_{H,B}) - \partial_m f(\mathbf{J}^{\pi_\theta}_H) - \nabla \partial_m f(\mathbf{J}^{\pi_\theta}_H)^\top (\widehat{\mathbf{J}}^{\pi_\theta}_{H,B} - \mathbf{J}^{\pi_\theta}_H)\right|$$
$$\le \frac{L_{2,f}}{2} \|\widehat{\mathbf{J}}^{\pi_\theta}_{H,B} - \mathbf{J}^{\pi_\theta}_H\|^2.$$

Taking the expectation and utilizing the fact that the empirical return estimate is unbiased, i.e., $\mathbb{E}[\widehat{\mathbf{J}}^{\pi_\theta}_{H,B}] = \mathbf{J}^{\pi_\theta}_H$, the linear term $\mathbb{E}[\nabla \partial_m f(\mathbf{J}^{\pi_\theta}_H)^\top (\widehat{\mathbf{J}}^{\pi_\theta}_{H,B} - \mathbf{J}^{\pi_\theta}_H)]$ vanishes. This yields:

$$\left|\mathbb{E}[\partial_m f(\widehat{\mathbf{J}}^{\pi_\theta}_{H,B})] - \partial_m f(\mathbf{J}^{\pi_\theta}_H)\right| \le \frac{L_{2,f}}{2} \mathbb{E}\|\widehat{\mathbf{J}}^{\pi_\theta}_{H,B} - \mathbf{J}^{\pi_\theta}_H\|^2. \tag{51}$$

By Lemma 4, we have $\mathbb{E}\|\widehat{\mathbf{J}}^{\pi_\theta}_{H,B} - \mathbf{J}^{\pi_\theta}_H\|^2 \le \frac{M}{(1-\gamma)^2 B}$. Substituting this into the inequality above, we obtain:

$$\left|\mathbb{E}[\partial_m f(\widehat{\mathbf{J}}^{\pi_\theta}_{H,B})] - \partial_m f(\mathbf{J}^{\pi_\theta}_H)\right| \le \mathcal{O}\left(\frac{L_{2,f} M}{(1-\gamma)^2 B}\right). \tag{52}$$

Applying this bound to Lemma 3(i) concludes the proof of Lemma 6(i).

For part (ii), condition on the return-estimation batch. The B_2 gradient trajectories are then independent, and each trajectory gradient has norm at most $CMG_1/(1-\gamma)^2$. Hence the conditional variance of their average is at most

$$\frac{C^2 G_1^2 M^2}{(1-\gamma)^4 B_2}.$$

By Jensen's inequality, Lipschitz continuity of $\partial_m f$, Lemma 4, and the horizon-truncation bound, the squared error of the conditional mean is at most

$$\frac{2L_f^2 G_1^2 M^3}{(1-\gamma)^6 B_1} + \mathcal{O}((1+H)^2\gamma^{2H}).$$

The conditional variance decomposition adds these two contributions and proves the MSE bound in Lemma 6.

J Proof of Theorems 1 and 2

J.1 Overall Proof Framework

All asymptotic statements below hold with problem parameters and the initial policy fixed. Assume that the initial expected KL divergence appearing below is finite.

Lemma 1 gives

$$\begin{aligned} f(\mathbf{J}^{\pi^*}) - \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[f(\mathbf{J}^{\pi_{\theta_k}})] \\ \le \frac{\sqrt{2\varepsilon_{\text{bias}}}}{1-\gamma} + \frac{G_1}{K}\sum_{k=0}^{K-1}\mathbb{E}\,\|\mathbb{E}[\omega_k \mid \theta_k] - \omega_k^*\| \\ + \frac{G_2\alpha}{2K}\sum_{k=0}^{K-1}\mathbb{E}\|\omega_k\|^2 + \frac{1}{\alpha K}\mathbb{E}_{s\sim d_\rho^{\pi^*}}\left[\mathrm{KL}\big(\pi^*(\cdot \mid s)\,\|\,\pi_{\theta_0}(\cdot \mid s)\big)\right]. \end{aligned} \tag{53}$$

By Assumption 4, $\|\omega_k^*\|^2 \le \mu^{-2}\|\nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})\|^2$. Hence

$$\|\omega_k\|^2 \le \frac{2}{\mu^2}\|\nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})\|^2 + 2\|\omega_k - \omega_k^*\|^2. \tag{54}$$

Substituting (54) into (53) gives

$$\begin{aligned} f(\mathbf{J}^{\pi^*}) - \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[f(\mathbf{J}^{\pi_{\theta_k}})] \\ \le \frac{\sqrt{2\varepsilon_{\text{bias}}}}{1-\gamma} + \frac{G_1}{K}\sum_{k=0}^{K-1}\mathbb{E}\,\|\mathbb{E}[\omega_k \mid \theta_k] - \omega_k^*\| \\ + \frac{G_2\alpha}{\mu^2 K}\sum_{k=0}^{K-1}\mathbb{E}\|\nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})\|^2 + \frac{G_2\alpha}{K}\sum_{k=0}^{K-1}\mathbb{E}\|\omega_k - \omega_k^*\|^2 \\ + \frac{1}{\alpha K}\mathbb{E}_{s\sim d_\rho^{\pi^*}}\left[\mathrm{KL}\big(\pi^*(\cdot \mid s)\,\|\,\pi_{\theta_0}(\cdot \mid s)\big)\right]. \end{aligned} \tag{55}$$

For $\alpha \le \mu^2/(4L_J G_1^2)$, Lemma 8 implies

$$\begin{aligned} \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|\nabla_\theta f(\mathbf{J}^{\pi_{\theta_k}})\|^2 \\ \le \frac{4G_1^2}{\alpha K}\mathbb{E}[f(\mathbf{J}^{\pi_{\theta_K}}) - f(\mathbf{J}^{\pi_{\theta_0}})] + \frac{2G_1^4}{K}\sum_{k=0}^{K-1}\mathbb{E}\,\|\mathbb{E}[\omega_k \mid \theta_k] - \omega_k^*\|^2 \\ + \frac{4G_1^2 L_J\alpha}{K}\sum_{k=0}^{K-1}\mathbb{E}\|\omega_k - \omega_k^*\|^2. \end{aligned} \tag{56}$$

Combining (56) with (55) yields

$$
\begin{aligned}
& f(\mathbf{J}^{\pi^*}) - \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[f(\mathbf{J}^{\pi_{\theta_k}})] \\
& \quad \le \frac{\sqrt{2\varepsilon_{\text{bias}}}}{1-\gamma} + \frac{G_1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left\|\mathbb{E}[\omega_k \mid \theta_k] - \omega_k^*\right\| \\
& \qquad + \frac{1}{\alpha K}\mathbb{E}_{s\sim d_\rho^{\pi^*}}\left[\mathrm{KL}\big(\pi^*(\cdot \mid s)\,\|\,\pi_{\theta_0}(\cdot \mid s)\big)\right] \\
& \qquad + \frac{4G_2G_1^2}{\mu^2 K}\mathbb{E}[f(\mathbf{J}^{\pi_{\theta_K}}) - f(\mathbf{J}^{\pi_{\theta_0}})] \\
& \qquad + \frac{2G_2G_1^4\alpha}{\mu^2 K}\sum_{k=0}^{K-1}\mathbb{E}\left\|\mathbb{E}[\omega_k \mid \theta_k] - \omega_k^*\right\|^2 \\
& \qquad + \left(\frac{G_2\alpha}{K} + \frac{4G_2G_1^2L_J\alpha^2}{\mu^2 K}\right)\sum_{k=0}^{K-1}\mathbb{E}\|\omega_k - \omega_k^*\|^2. \qquad (57)
\end{aligned}
$$

We now apply the error bounds in Lemma 2 to each estimator.

J.2 Proof of Theorem 1

For Algorithm 2, set

$$
\alpha = \min\left\{1, \frac{\mu^2}{4L_JG_1^2}\right\}\frac{\epsilon}{1+\log(1/\epsilon)}, \qquad B_{\max} = \epsilon^{-2}, \qquad B = 1, \qquad K = \left\lceil \frac{1}{\alpha\epsilon}\right\rceil.
$$

Initialize every inner loop at $\omega_0^k = 0$ and choose

$$
\begin{aligned}
\beta &= \frac{\mu}{4G_1^4}, \\
H &= \left\lceil \frac{\log\big(\max\{\epsilon^{-2}, 2G_1^2/\mu\}\big)}{\log(1/\gamma)}\right\rceil, \qquad (58) \\
N &= \left\lceil \frac{32G_1^4}{\mu^2}\log(1/\epsilon)\right\rceil.
\end{aligned}
$$

For $0 < \epsilon < 1$, these choices imply

$$
\gamma^H \le \min\{\epsilon^2, \mu/(2G_1^2)\}, \qquad \gamma^{2H} \le \epsilon^4, \qquad (59)
$$

and

$$
\exp\left(-\frac{N\mu^2}{16G_1^4}\right) \le \epsilon^2. \qquad (60)
$$

The initial inner-loop error is uniformly bounded: Assumption 4 gives $\|\omega_0^k - \omega_k^*\| \le CMG_1/(\mu(1-\gamma)^2)$. Thus (60) also controls the initial-error term in Lemma 2 to $\mathcal{O}(\epsilon^2)$.

Lemma 5, including trajectory noise and horizon truncation, gives

$$
\bar{\delta}_g = \mathcal{O}(\epsilon), \qquad \sigma_g^2 = \mathcal{O}(1+\log(1/\epsilon)).
$$

Applying Lemma 2 with $B = 1$ and using (59)–(60) therefore yields

$$
\begin{aligned}
\mathbb{E}\|\omega_k - \omega_k^*\|^2 &= \mathcal{O}(1+\log(1/\epsilon)), \qquad (61) \\
\mathbb{E}\left\|\mathbb{E}[\omega_k \mid \theta_k] - \omega_k^*\right\| &= \mathcal{O}(\epsilon), \qquad (62) \\
\mathbb{E}\left\|\mathbb{E}[\omega_k \mid \theta_k] - \omega_k^*\right\|^2 &= \mathcal{O}(\epsilon^2).
\end{aligned}
$$

These bounds hold uniformly in k.

By (62),

$$\frac{G_1}{K}\sum_{k=0}^{K-1}\mathbb{E}\,\|\mathbb{E}[\omega_k \mid \theta_k]-\omega_k^*\| = \mathcal{O}(\epsilon), \tag{63}$$

$$\frac{2G_2G_1^4\alpha}{\mu^2 K}\sum_{k=0}^{K-1}\mathbb{E}\,\|\mathbb{E}[\omega_k \mid \theta_k]-\omega_k^*\|^2 = \mathcal{O}(\alpha\epsilon^2). \tag{64}$$

By (61) and $\alpha \le 1$,

$$\left(\frac{G_2\alpha}{K}+\frac{4G_2G_1^2L_J\alpha^2}{\mu^2 K}\right)\sum_{k=0}^{K-1}\mathbb{E}\|\omega_k-\omega_k^*\|^2 = \mathcal{O}\big(\alpha(1+\log(1/\epsilon))\big) = \mathcal{O}(\epsilon). \tag{65}$$

The range of $f(\mathbf{J}^{\pi_\theta})$ is at most $MC/(1-\gamma)$. Consequently,

$$\frac{4G_2G_1^2}{\mu^2 K}\mathbb{E}[f(\mathbf{J}^{\pi_{\theta_K}})-f(\mathbf{J}^{\pi_{\theta_0}})] = \mathcal{O}(1/K). \tag{66}$$

The finite initial KL term in (57) is $\mathcal{O}(1/(\alpha K)) = \mathcal{O}(\epsilon)$. Combining (63), (64), (65), and (66) with (57) gives

$$\begin{aligned} f(\mathbf{J}^{\pi^*}) &- \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[f(\mathbf{J}^{\pi_{\theta_k}})] \\ &\le \frac{\sqrt{2\varepsilon_{\text{bias}}}}{1-\gamma} + \mathcal{O}\left(\epsilon+\alpha\epsilon^2+\alpha(1+\log(1/\epsilon))+\frac{1}{\alpha K}+\frac{1}{K}\right) \\ &= \frac{\sqrt{2\varepsilon_{\text{bias}}}}{1-\gamma} + \mathcal{O}(\epsilon). \end{aligned} \tag{67}$$

This proves the performance guarantee in Theorem 1.

Each outer iteration uses the random return batch and NB inner-loop trajectories. The same inner-loop batch estimates both Fisher and gradient and is counted once. With the base trajectory reused, the expected total trajectory count is

$$K\left(\lfloor\log_2 B_{\max}\rfloor + 2^{-\lfloor\log_2 B_{\max}\rfloor} + NB\right) = \widetilde{\mathcal{O}}(\epsilon^{-2}).$$

Multiplying by H gives the same $\widetilde{\mathcal{O}}(\epsilon^{-2})$ expected transition count.

J.3 Proof of Theorem 2

For Algorithm 1, set

$$\alpha = \min\left\{1, \frac{\mu^2}{4L_JG_1^2}\right\}, \qquad B_1 = B_2 = B = \left\lceil\frac{1}{\epsilon(1-\gamma)^2}\right\rceil, \qquad K = \left\lceil\frac{1}{\alpha\epsilon}\right\rceil.$$

Use the same H, N, β and zero initialization as in (58). In particular,

$$\gamma^H \le \min\{\epsilon^2, \mu/(2G_1^2)\}, \qquad \gamma^{2H} \le \epsilon^4, \tag{68}$$

$$\exp\left(-\frac{N\mu^2}{16G_1^4}\right) \le \epsilon^2. \tag{69}$$

The batched-estimator bounds in Lemma 6 give

$$\bar{\delta}_g = \mathcal{O}(\epsilon), \qquad \sigma_g^2 = \mathcal{O}(\epsilon).$$

Applying Lemma 2 with the inner batch $B_2 = B$, and using (68) and (69), yields

$$\mathbb{E}\|\omega_k-\omega_k^*\|^2 = \mathcal{O}(\epsilon), \tag{70}$$

$$\mathbb{E}\,\|\mathbb{E}[\omega_k \mid \theta_k]-\omega_k^*\| = \mathcal{O}(\epsilon), \tag{71}$$

$$\mathbb{E}\,\|\mathbb{E}[\omega_k \mid \theta_k]-\omega_k^*\|^2 = \mathcal{O}(\epsilon^2).$$

Substituting (70) and (71) into (57), the mean-bias term is $\mathcal{O}(\epsilon)$, the squared-bias term is $\mathcal{O}(\epsilon^2)$, and the MSE term is $\mathcal{O}(\epsilon)$. The KL term is $\mathcal{O}(1/(\alpha K)) = \mathcal{O}(\epsilon)$, and the objective-range term is $\mathcal{O}(1/K) = \mathcal{O}(\epsilon)$. Therefore

$$f(\mathbf{J}^{\pi^*}) - \frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}[f(\mathbf{J}^{\pi_{\theta_k}})] \leq \frac{\sqrt{2\varepsilon_{\text{bias}}}}{1-\gamma} + \mathcal{O}(\epsilon),$$

as claimed in Theorem 2.

The return batch is sampled once per outer iteration and reused throughout its inner loop. Thus the trajectory count is

$$K(B_1 + NB_2) = KB(1+N) = \widetilde{\mathcal{O}}(\epsilon^{-2}).$$

Multiplying by H gives the same $\widetilde{\mathcal{O}}(\epsilon^{-2})$ transition count.

J.4 Empirical NPG under Assumptions 1–5

Under Assumptions 1–5, Algorithm 1 also admits a $\widetilde{\mathcal{O}}(\epsilon^{-3})$ sample-complexity guarantee. To see this, use the common initialization, β, H, N, and the constant outer step size from Theorem 2, with

$$B_1 = \left\lceil \frac{1}{\epsilon^2(1-\gamma)^2} \right\rceil, \qquad B_2 = \left\lceil \frac{1}{\epsilon(1-\gamma)^2} \right\rceil, \qquad K = \left\lceil \frac{1}{\alpha\epsilon} \right\rceil.$$

Lipschitz continuity gives

$$\bar{\delta}_g = \mathcal{O}(B_1^{-1/2} + (1+H)\gamma^H) = \mathcal{O}(\epsilon).$$

The conditional-variance argument for the empirical estimator does not require Assumption 6 and gives

$$\sigma_g^2 = \mathcal{O}(B_2^{-1} + B_1^{-1} + (1+H)^2\gamma^{2H}) = \mathcal{O}(\epsilon).$$

Lemma 2 and (57) therefore give an $\mathcal{O}(\epsilon)$ performance error above $\sqrt{2\varepsilon_{\text{bias}}}/(1-\gamma)$. The trajectory count is

$$K(B_1 + NB_2) = \widetilde{\mathcal{O}}(\epsilon^{-3}),$$

and multiplication by H preserves this accuracy exponent for the transition count.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline our primary contributions regarding sample complexity bounds for MORL. We define the problem setting in the introduction and provide pointers to the specific technical assumptions detailed in Section 4.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We acknowledge that our results rely on the assumptions stated in Section 4. In the conclusion, we also note that our guarantees do not directly extend to RLGU with function approximation.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The full set of assumptions used can be found in Section 4. In the same section, the proof outline can be found where all supporting details are included in the Appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The pseudocode of the algorithm with all details is provided in the main paper (Section 3) and code for the experiments is publicly available at `https://github.itap.purdue.edu/Clan-labs/MLMC-NPG-ConcaveMORL` (Appendix B).

Guidelines:

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

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The code is publicly available at `https://github.itap.purdue.edu/Clan-labs/MLMC-NPG-ConcaveMORL` (Appendix B).

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide such experimental details in Appendix B.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: All experimental plots display the mean performance aggregated over 20 independent random seeds, with shaded regions representing one standard deviation. These details are provided in Appendix B.

   Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Yes. We have reported that all experiments were run on a small local workstation with an AMD Ryzen 5 3600X processor (6 cores), 32 GB RAM, and an NVIDIA GeForce RTX 3080 Ti GPU with 12 GB VRAM.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conducted in the paper conform with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This work focuses on the theoretical foundations of Multi-Objective Reinforcement Learning (MORL). Thus, it does not have a direct application to specific real-world deployments and there are no immediate foreseeable negative societal impacts. In the long

term, improvements in the theoretical understanding of MORL may lead to more robust and transparent decision-making systems in complex environments.

Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All code packages or datasets used are cited appropriately.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The code is released at `https://github.itap.purdue.edu/Clan-labs/MLMC-NPG-ConcaveMORL` (Appendix B).

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

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

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were only used for editing purpose and were not used to generate original components.

    Guidelines:

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