ONE GRADIENT, YET MORE ROBUST POLICIES: MOMENTUM-GUIDED POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Robust reinforcement learning (Robust RL) aims to learn policies that remain effective under environmental uncertainty. Recent methods pursue such robustness by seeking flat reward maxima, often through sharpness-aware minimization (SAM), which encourages high returns under small parameter perturbations. However, vanilla SAM first constructs an adversarial parameter perturbation from a stochastic policy gradient and then computes a second gradient at the perturbed parameters to update the policy. This procedure relies on noisy gradient estimates and incurs additional computation, which can make it complex and unstable. In this paper, we introduce a policy optimization method, based on Momentum-SAM (mSAM), for Robust RL that utilizes accumulated momentum to construct adversarial perturbations. The resulting method is effective and computationally lightweight, requiring only one policy-gradient evaluation per update while reducing reliance on a single noisy estimate. Furthermore, we establish finite-time stationarity bounds for SAM and mSAM in policy optimization, providing a theoretical foundation for sharpness-aware policy learning. Our experiments show the effectiveness of mSAM in enhancing the robustness of RL compared with existing methods under diverse perturbations, including actions, transition dynamics, and reward functions.

1 INTRODUCTION

Progress in reinforcement learning (RL) has enabled agents to tackle increasingly complex tasks, with notable successes in games and robotics (Silver et al., 2016; Jaderberg et al., 2019; Tang et al., 2025). Despite these successes, RL remains vulnerable to small perturbations in real-world environments, which can substantially degrade the performance of learned policies (Tessler et al., 2019). For example, a robot trained under nominal conditions may struggle to maintain stable behavior when it carries a heavier load or encounters a surface with lower friction. Robust reinforcement learning (Robust RL, Morimoto & Doya, 2005) addresses this issue by learning policies that remain effective under worst-case environmental uncertainty, including robust policy optimization (Rahman & Xue, 2022) and action robust RL (Tessler et al., 2019). However, these methods often require task-specific designs and complex training procedures (Pinto et al., 2017; Zhang et al., 2023), which can limit their applicability to a broader range of RL tasks.

Recently, adversarial perturbations in policy-parameter space have been explored as an alternative approach to policy robustness (Lee et al., 2023; Ramasubramanian et al., 2025; Lee & Yoon, 2025). The underlying idea is to seek flat reward maxima (Sullivan et al., 2022), where policies maintain high returns even under unfavorable perturbations within a local parameter neighborhood. Such flatness has been linked to action robustness and improved generalization to environments with perturbed dynamics (Lee & Yoon, 2025; Mani et al., 2026). Sharpness-aware minimization (SAM, Foret et al., 2021) pursues this goal through a min-max formulation that minimizes the worst-case loss within a bounded parameter neighborhood.

However, constructing adversarial perturbations for SAM poses two challenges in policy optimization. 1) Standard SAM requires one gradient evaluation to construct the perturbation and another at the perturbed parameters to update the policy (Li et al., 2024), increasing training cost and limiting its scalability (see Appendix A.2 for details). 2) Perturbations derived from a single mini-batch gradient can be sensitive to sampling noise and may poorly represent the sharpness of the underlying objective (Li & Giannakis, 2023). In RL, high-variance policy-gradient estimates can further undermine the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

reliability of these perturbations in capturing local sharpness (Ilyas et al., 2020). This raises a natural question: *Can we make sharpness-aware policy optimization more reliable and efficient?*

To this end, we propose mSAM for policy optimization, using historical policy gradients to guide adversarial perturbations. Inspired by Nesterov-style lookahead in adversarial perturbation search (Korkmaz, 2020) and Momentum-SAM (Becker et al., 2025), we construct adversarial perturbations in the direction of the accumulated momentum vector to achieve flat reward without significant computational overhead. Accumulated gradients from previous updates guide perturbation selection, reducing reliance on a single noisy batch. The policy update then uses the gradient at the perturbed parameters, avoiding a separate gradient evaluation for perturbation construction. We further establish finite-time stationarity guarantees for both SAM and mSAM in policy optimization, providing a theoretical basis for these methods and quantifying the optimization error introduced by parameter perturbations. Experiments on three MuJoCo benchmarks, Hopper, Walker2d, and HalfCheetah, show that mSAM improves robustness under action and dynamics perturbations during evaluation and maintains effective policy learning with noisy training rewards.

Contributions. We summarize our main contributions as follows:

- We introduce mSAM to policy optimization through momentum-guided adversarial perturbations. The resulting method improves policy robustness without the extra gradient cost of SAM.
- We establish finite-time stationarity bounds for SAM and mSAM policy optimization, providing a theoretical foundation for sharpness-aware policy learning. Our analysis quantifies how gradient noise and parameter perturbations affect the resulting optimization guarantees.
- We demonstrate the effectiveness of mSAM on several tasks under action perturbations, dynamics variations, and noisy training rewards. The results demonstrate the effectiveness and computational efficiency of mSAM.

1.1 Related Work

We briefly review the main related work here and refer readers to Appendix B for a broader discussion of robust RL, momentum in RL, and efficient variants of SAM.

Robust RL aims to learn policies that remain effective under uncertainty and perturbations. Classical robust Markov decision processes formalize this objective through uncertainty sets over transition models and robust dynamics (Nilim & El Ghaoui, 2005). In deep RL, adversarial formulations have been developed for environmental disturbances (Pinto et al., 2017) and action perturbations (Tessler et al., 2019). These approaches primarily seek robustness to changes in environment dynamics or executed actions.

Another line of work uses SAM to reduce sensitivity to parameter perturbations. In RL, SAM has been applied to preserve plasticity under non-stationary training data (Lee et al., 2023) and improve locomotion robustness (Bochem et al., 2024). Lee & Yoon (2025) further establish a connection between reward flatness and action robustness as well as propose SAM-enhanced PPO. Other studies explore sharpness-aware learning with corrupted offline data (Xu & Chen, 2026), stabilize sharpness-aware updates under bootstrapped targets (Wang et al., 2026), and use parameter sensitivity as an uncertainty proxy for safe exploration (Mani et al., 2026).

Despite these advances, the computational overhead and gradient noise sensitivity of standard SAM remain challenges in policy optimization. To this end, we develop an efficient sharpness-aware policy optimization algorithm and further establish its theoretical guarantees.

2 Preliminaries

In this section, we review the necessary background for our method: 1) Markov Decision Processes (MDPs); 2) robust MDPs; and 3) Sharpness-Aware Minimization (SAM), an optimization method for finding flatter minima.

Markov Decision Processes (MDPs). We consider a discounted MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, P represents the state transition kernel, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$

is the reward function, and $\gamma \in (0, 1)$ is the discount factor. At each timestep t, an agent observes a state $s_t \in \mathcal{S}$, selects an action $a_t \sim \pi(\cdot \mid s_t)$, receives a reward $r(s_t, a_t)$, and transitions to the next state $s_{t+1} \sim P(\cdot \mid s_t, a_t)$. The goal is to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the expected discounted return,

$$V^{\pi}(s) := \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s\Big], \tag{1}$$

where the expectation is over trajectories generated by following π under the transition kernel P.

Policy gradient methods (Williams, 1992; Sutton et al., 1999) are widely used in deep RL to optimize this objective directly. Specifically, the policy is represented by a neural network $\pi_{\boldsymbol{\theta}}(a|s)$ with learnable parameters $\boldsymbol{\theta}$. Then, the gradient of the expected return with respect to $\boldsymbol{\theta}$ can be given by: $\nabla_{\boldsymbol{\theta}} V^{\pi_{\boldsymbol{\theta}}}(s) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t)\, G_t | s_0 = s\Big]$, where $G_t = \sum_{k=t}^{\infty} \gamma^{k-t} r(s_k, a_k)$ denotes the discounted return from time t onwards.

Robust MDPs. Robust MDPs seek policies that maintain performance under uncertainty in the environment dynamics (Iyengar, 2005; Nilim & El Ghaoui, 2005). Specifically, rather than assuming a single known transition kernel, a robust MDP specifies an uncertainty set $\mathcal{P}$ of plausible transition kernels and optimizes the worst-case expected discounted return:

$$\max_{\pi} \min_{P \in \mathcal{P}} \mathbb{E}_{P,\pi}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\Big], \tag{2}$$

The inner minimization in Eq. (2) evaluates each policy under its least favorable dynamics within $\mathcal{P}$, while the outer maximization selects a policy that achieves the highest worst-case return.

Flat Minima and Sharpness-Aware Minimization (SAM). Recent studies (Jiang et al., 2019; Andriushchenko & Flammarion, 2022) have indicated that the sharpness of the loss landscape correlates strongly with generalization, where flatter minima tend to generalize better. Building on these observations, SAM (Foret et al., 2021) and recent variants (Kim et al., 2022; Oikonomou & Loizou, 2025) aim to seek flat minima by solving the following minimax optimization problem:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \max_{\|\boldsymbol{\epsilon}\| \le \rho} \mathcal{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon}), \tag{3}$$

where $\mathcal{L}(\boldsymbol{\theta})$ is the loss function, $\boldsymbol{\epsilon}$ is a parameter perturbation constrained to an ℓ_2 ball of radius ρ. In practice, SAM solves the above min-max problem within two phases: 1) approximates the perturbation as $\boldsymbol{\epsilon} = \rho \nabla \mathcal{L}(\boldsymbol{\theta}) / \|\nabla \mathcal{L}(\boldsymbol{\theta})\|$; 2) updates the model with the gradient at perturbed parameters $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla \mathcal{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon})$, where η is the learning rate.

3 Method

In this section, we present SAM for policy optimization, introduce momentum-guided single-pass updates and their implementation in PPO, and discuss the updates from a local curvature perspective.

3.1 SAM for Policy Optimization

In the context of policy optimization, SAM framework (Foret et al., 2021) suggests to look for a policy π that minimizes

$$\min_{\boldsymbol{\theta}} \max_{\|\boldsymbol{\epsilon}\| \le \rho} L(\boldsymbol{\theta} + \boldsymbol{\epsilon}), \quad \text{where } L(\boldsymbol{\theta} + \boldsymbol{\epsilon}) = \mathbb{E}_{p, \pi_{\boldsymbol{\theta}+\boldsymbol{\epsilon}}}\Big[\sum_{t=0}^{\infty} -\gamma^t r(s_t, a_t)\Big]. \tag{4}$$

where ρ controls the perturbation radius. In Eq. (4), the inner maximization identifies the worst loss within this neighborhood, while the outer minimization seeks a policy $\pi_{\boldsymbol{\theta}}$ with high worst-case local return $V^{\pi_{\boldsymbol{\theta}+\boldsymbol{\epsilon}}}$. Eq. (4) promotes flat reward maxima by encouraging stable returns under local parameter perturbations. Under specific assumptions, such flatness is also linked to action robustness (Lee & Yoon, 2025). Further discussion of this connection is provided in Appendix A.4.

3.2 Momentum-Guided Single-Pass Updates

Despite its ability to promote flat rewards, standard SAM suffers from noisy sharpness estimates and substantial computational overhead, hindering its efficient application to RL. First, the perturbation $\boldsymbol{\epsilon}$

relies on gradients estimated from finite trajectory batches, whose sampling noise can destabilize its direction and compromise sharpness estimation (Li & Giannakis, 2023). Second, constructing $\boldsymbol{\epsilon}$ and computing the update gradient at $\boldsymbol{\theta} + \boldsymbol{\epsilon}$ require two sequential gradient evaluations, adding substantial computational overhead. These limitations highlight the need for a more efficient way to incorporate sharpness awareness into policy learning.

Inspired by Momentum-SAM (Becker et al., 2025), we construct the perturbation from accumulated momentum as $\boldsymbol{\epsilon} = -\rho \boldsymbol{v}/\|\boldsymbol{v}\|$. The resulting update takes the form,

$$\boldsymbol{v}_{t+1} = \beta \boldsymbol{v}_t + \nabla L\big(\boldsymbol{\theta} - \rho \boldsymbol{v}_t/\|\boldsymbol{v}_t\|\big), \quad \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \boldsymbol{v}_{t+1}, \tag{5}$$

where β is the momentum coefficient. This construction reuses the stored momentum to determine the perturbation and requires only one new gradient evaluation at $\boldsymbol{\theta} + \boldsymbol{\epsilon}$.

Accumulated momentum combines information across previous updates, which can smooth mini-batch fluctuations in the perturbation direction. The negative direction is motivated by MSAM's observation that momentum can overshoot a local minimum, allowing a further displacement along the same direction to expose increased loss (Becker et al., 2025). Since the momentum is already available, constructing this perturbation requires no additional gradient evaluation.

Practical Algorithm. Using proximal policy optimization (PPO) as an example, we describe how mSAM can be incorporated into policy learning. At each rollout iteration, PPO+mSAM 1) collects trajectories using the current policy; 2) computes advantage estimates and value targets; 3) constructs the negative PPO clipped loss, keeping the behavior-policy probabilities and advantages fixed; 4) forms the momentum-based perturbation $\boldsymbol{\epsilon} = -\rho \boldsymbol{v}/\|\boldsymbol{v}\|$ and evaluates the policy-loss gradient at $\boldsymbol{\theta} + \boldsymbol{\epsilon}$; 5) restores the nominal parameters $\boldsymbol{\theta}$ and uses this gradient to update the momentum and policy; and 6) fits the value function to the fixed targets by minimizing the mean-squared error. Detailed pseudocode is provided in Algorithm 2.

3.3 Discuss

We now discuss how momentum-guided perturbations influence policy learning, connecting their local geometric interpretation to changes in policy behavior and the evolving role of historical gradients. For a locally twice-differentiable loss and sufficiently small ρ,

$$\nabla L(\boldsymbol{\theta} + \boldsymbol{\epsilon}) \approx \nabla L(\boldsymbol{\theta}) - \rho \nabla^2 L(\boldsymbol{\theta}) \widehat{\boldsymbol{v}}$$

where $\widehat{\boldsymbol{v}} = \boldsymbol{v}/|\boldsymbol{v}|$. Thus, the perturbation allows local curvature along the momentum direction to influence the update without explicitly computing the Hessian.

The choice of this direction is motivated by (Becker et al., 2025), who observe that momentum directions can exhibit high curvature and that further movement along negative momentum after overshooting a local minimum may increase the loss. This suggests that historical gradients can help identify nearby loss-sensitive regions, although the resulting perturbation need not maximize the local loss.

In RL, this gradient correction can also influence policy behavior through the resulting parameter update. To first order,

$$\pi_{\boldsymbol{\theta}+\Delta\boldsymbol{\theta}}(a \mid s) - \pi_{\boldsymbol{\theta}}(a \mid s) \approx \nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(a \mid s)^\top \Delta\boldsymbol{\theta}.$$

Since actions influence state transitions, changes in the policy's action distribution can also alter the states encountered during subsequent interaction. Hence, gradients accumulated under earlier policies may become less representative of the current learning conditions. Momentum therefore offers a balance between smoothing sampling fluctuations and responding to changes during policy learning.

4 Theoretical Results

In this section, we provide a theoretical analysis of SAM-based policy optimization in robust RL. While existing studies have primarily focused on algorithmic design, the theoretical understanding of SAM in this setting remains limited. We first derive a finite-time stationarity guarantee for SAM (Theorem 1) and then establish the convergence of our proposed mSAM (Theorem 3). Due to space constraints, the assumptions and proofs are deferred to Appendix E.

Theorem 1 (Finite-time stationarity of SAM policy optimization)**.** *Under Algorithm 1, suppose Assumptions 1 and 2 hold. If* $\rho_t = \rho \le \min\left\{\frac{1}{4L}, \frac{1}{8BL(1-\alpha)^2}\right\}, \eta_t = \eta \le \frac{1}{8BL}$*, then, for every integer* $T \ge 1$*, the iterates satisfy*

$$\min_{0\le t<T} \mathbb{E}[\|\nabla V(\boldsymbol{\theta}_t, \xi)\|^2] \le \frac{\Delta(\boldsymbol{\theta}_0)}{T\eta} + 2LN_{\eta,\rho,\alpha}, \tag{6}$$

where $N_{\eta,\rho,\alpha}$ *are finite constants that depending on* η, ρ, α*. The detailed expression can be found in Appendix E.1.*

This result shows that SAM policy optimization can converge to the first-order stationary point. When the SAM perturbation is interpreted as a local adversarial perturbation of the policy parameters, the preceding result provides an approximate-stationary guarantee for learning locally robust policies. In particular, a fixed nonzero radius generally produces a non-vanishing stationarity neighborhood. For normalized SAM, corresponding to $\alpha = 1$, this neighborhood contains an $O(\rho)$ term even in the exact-gradient case. This reflects the optimization cost of maintaining robustness against perturbations of a fixed magnitude. Exact stationarity requires the perturbation radius to vanish, or alternatively requires increasing batch sizes or variance reduction when stochastic-gradient noise is present.

With different settings of step size η_t and ρ_t, we can obtain the feasible convergence rate. We will discuss this in Appendix D.4. Next, we give the convergence guarantee of the proposed mSAM policy optimization.

Theorem 2. *Under Assumption 1 and Algorithm 2, if the step-size satisfies* $\eta_t \le \frac{(1-\beta)^3}{L(\beta^2+2(1-\beta)^2)}$ *and the gradients norm satisfies* $\mathbb{E}\left[\|\nabla V(\boldsymbol{\theta};\xi_t)\|^2 \mid \mathcal{F}_t\right] \le \|\nabla V(\boldsymbol{\theta})\|^2 + \frac{\sigma^2}{B}$*, then we have*

$$\min_t \mathbb{E}\big[\|\nabla V(\boldsymbol{\theta}_t, \xi)\|^2\big] \lesssim \underbrace{\frac{\Delta(\boldsymbol{\theta}_0)}{\sum_{t=1}^T \eta_t}}_{\text{optimization error}} + \underbrace{\frac{\sigma^2 \sum_{t=1}^T \eta_t^2}{B\sum_{t=1}^T \eta_t}}_{\text{stochastic-gradient error}} + \underbrace{\frac{\sum_{t=1}^T \eta_t \rho_t^2}{\sum_{t=1}^T \eta_t}}_{\text{sharpness-perturbation error}}, \tag{7}$$

where $B = |\mathcal{B}_t|$ *is the batch size, and* σ^2 *is the gradients variance.*

Theorem 2 is a simple case of Theorem 3, which shows that the proposed method converges to the first-order stationary point, and the errors come from three parts. The first term is the *optimization error*, which reflects the effect of the initial objective gap. Its decay is governed by the cumulative step size $\sum_{t=1}^T \eta_t$. The second term represents the *stochastic-gradient error*. It arises from the randomness of the stochastic gradient. This term exhibits the usual bias-variance trade-off in stochastic optimization. Larger step sizes and larger batch size B promote faster descent. The third term is specific to the sharpness-aware perturbation mechanism and quantifies the additional error induced by evaluating the stochastic gradient at the perturbed point. Since $\|\boldsymbol{z}_t - \boldsymbol{\theta}_t\| = \rho_t$, the smoothness of the objective function converts the discrepancy between gradients at these points into an error of order ρ_t^2, which accumulates through the weighted quantity $\sum_{t=1}^T \eta_t \rho_t^2$.

Remark. *Theorem 3 not only provides a convergence guarantee for the proposed mSAM policy optimization algorithm but also applies to general nonconvex optimization problems. Compared with ordinary momentum SAM methods, our algorithm has lower computation and storage costs and higher efficiency, since we do not need to compute the gradient vectors or momentum vectors twice in one round.*

5 Experiments

In this section, we evaluate the robustness of our method to action noise, transition perturbations, and noisy training rewards in continuous-control tasks. Additional results on discrete-action tasks, other policy optimization algorithms, and reward landscape flatness are provided in Appendix G.

5.1 Experimental Setup

Environments. We evaluate our method on three continuous control tasks from MuJoCo (Todorov et al., 2012): *HalfCheetah-v4*, *Hopper-v4*, and *Walker2d-v4*. In *HalfCheetah-v4*, the agent controls a

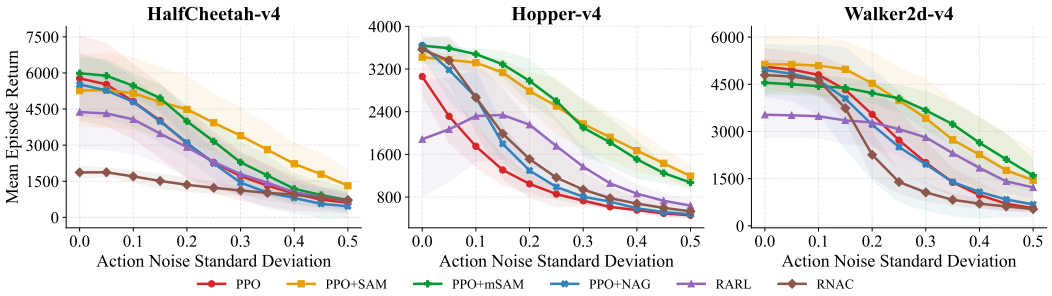


Figure 1: **Action robustness across different environments.** The average return is plotted against the action-noise standard deviation σ_a. We observe that PPO+MSAM achieves higher returns than PPO, PPO+NAG, RARL, and RNAC at the tested noise levels $\sigma_a \geq 0.2$ across all three environments. PPO+MSAM is competitive with PPO+SAM at low noise levels, whereas PPO+SAM generally retains higher returns as the noise increases, particularly in Walker2d-v4.

two-legged robot to run forward; in *Hopper-v4*, it controls a one-legged robot to hop forward while maintaining balance; in *Walker2d-v4*, it controls a bipedal robot to walk forward without falling. Across all three tasks, the objective is to maximize forward progress while limiting control effort.

Baselines. We consider two categories of baselines: 1) robust reinforcement learning methods, including RNAC (Zhou et al., 2023) and RARL (Pinto et al., 2017); and 2) PPO variants with alternative optimization methods, including PPO+SAM (Lee & Yoon, 2025) and PPO+NAG, which integrates Nesterov's accelerated gradient (Chen et al., 2023) into PPO optimization.

Implementation details. To ensure a fair comparison among baseline methods, we follow the settings of Lee & Yoon (2025). We repeat each experiment with five seeds and average the results over 100 runs per evaluation to reduce variability. Detailed settings are provided in Appendix F.

5.2 Robustness Evaluation under Perturbations

Robustness under Action Perturbations. To assess whether learned policies remain effective under noisy action execution, we evaluate them across different environments with additive Gaussian action noise. During evaluation, we add zero-mean Gaussian noise to the policy actions $a' = a + \mathcal{N}(0, \sigma_a^2)$. We vary the noise standard deviation σ_a from 0 to 0.5 and report the average episodic return at each noise level. The results are presented in Fig. 1. PPO+MSAM maintains higher returns than PPO and PPO+NAG under moderate-to-high noise across all three environments, demonstrating the benefit of incorporating momentum-based perturbations into policy optimization.

Robustness under Transition Perturbations. To evaluate whether the learned policies generalize to changes in the agent's physical properties, we perturb the transition dynamics by modifying two physical parameters of the environment: ground friction and torso mass.

We first vary the friction coefficient between the agent's feet and the ground using scaling factors from 0.4 to 1.6, with 1.0 corresponding to the nominal setting, while keeping torso mass fixed. This simulates different surface conditions, ranging from slippery to rough terrain, and evaluates whether the policies can maintain effective locomotion and balance under changes in ground friction. Fig. 2 presents the average episode returns across environments. PPO+mSAM generally achieves higher returns than PPO across the evaluated friction conditions, suggesting improved robustness to changes in surface properties, although its advantage over PPO+SAM varies across environments.

Then, we vary the torso mass scaling factor from 0.5 to 1.5, with 1.0 corresponding to the nominal mass, while keeping friction fixed. This perturbation alters the agent's dynamics and evaluates whether the policies can maintain effective locomotion and balance under changes in body mass. Fig. 3 presents the average episode returns across three environments. PPO+mSAM improves performance over PPO across a broad range of mass factors, particularly on HalfCheetah-v4 and under increased torso mass on Hopper-v4, while its advantage over PPO+SAM remains environment-dependent.

Finally, we jointly vary the torso mass scaling factor from 0.5 to 1.5 and the friction scaling factor from 0.4 to 1.6, with $(1.0, 1.0)$ corresponding to the nominal setting. This evaluation assesses whether the learned policies can maintain effective locomotion under simultaneous changes in body mass and ground conditions. Fig. 4 presents heatmaps of the average episode returns across the three environments. Overall, PPO+mSAM maintains strong performance over a broader range of combined perturbations than PPO, particularly on HalfCheetah-v4, although its robustness remains environment-dependent and substantial sensitivity to friction persists on Hopper-v4.

Robustness under Reward Perturbations. To evaluate whether effective policies can be learned from noisy reward signals, we add zero-mean Gaussian noise to training rewards with standard deviation $\sigma_r \in \{0.0, 0.1, 0.2, 0.5, 1.0\}$ and evaluate the resulting policies in the nominal environment without reward noise. Table 1 reports the mean episodic returns over five seeds. PPO+mSAM achieves the highest mean return in 11 of the 15 settings and outperforms all baselines in all three environments at the highest noise level. Performance does not decline monotonically with increasing noise: some nonzero noise levels yield higher mean returns than the noise-free setting. These results support the effectiveness of PPO+mSAM in learning from noisy rewards.

Table 1: **Robustness evaluation under training reward noise across different environments.** PPO+mSAM achieves the highest mean return in 11 of the 15 settings. The best and second-best results are bold and underlined, respectively.

	Method	$\sigma_r = 0.0$	$\sigma_r = 0.1$	$\sigma_r = 0.2$	$\sigma_r = 0.5$	$\sigma_r = 1.0$
HalfCheetah	PPO	<u>5775.4</u> ± 1786.5	5949.7 ± 947.2	5046.0 ± 1414.0	5453.1 ± 880.9	5162.0 ± 804.9
	PPO+SAM	5287.1 ± 1545.4	<u>5960.3</u> ± 768.0	5530.6 ± 2315.9	<u>6403.5</u> ± 1184.2	<u>5557.4</u> ± 1311.8
	PPO+mSAM	**6218.3** ± 1074.9	**6742.0** ± 1549.4	**6652.8** ± 1010.6	6372.3 ± 934.1	**5969.1** ± 1301.2
	PPO+NAG	5487.3 ± 1290.3	4886.2 ± 1743.5	<u>6187.7</u> ± 947.8	**6449.5** ± 679.8	5156.7 ± 1880.9
	RARL	4348.9 ± 1594.0	3675.2 ± 561.6	4145.3 ± 908.1	4038.7 ± 701.0	4950.5 ± 1662.5
	RNAC	1893.1 ± 228.7	2011.1 ± 105.5	3399.9 ± 2254.1	2833.2 ± 1966.5	4844.2 ± 1841.3
Hopper	PPO	3091.1 ± 310.4	3278.2 ± 666.0	3513.7 ± 226.6	3051.2 ± 1131.5	3303.6 ± 125.3
	PPO+SAM	3512.5 ± 179.9	3454.1 ± 336.8	3404.0 ± 240.1	3481.8 ± 194.9	<u>3548.8</u> ± 224.9
	PPO+mSAM	**3639.4** ± 37.5	**3539.4** ± 95.5	**3586.6** ± 161.5	<u>3503.7</u> ± 318.8	**3613.8** ± 62.8
	PPO+NAG	<u>3624.3</u> ± 190.0	<u>3527.2</u> ± 118.6	3497.4 ± 303.4	3477.8 ± 376.5	3481.9 ± 111.0
	RARL	1868.1 ± 1094.5	2520.0 ± 1071.8	2369.3 ± 1230.5	2436.9 ± 1166.0	2643.8 ± 1145.7
	RNAC	3560.1 ± 220.0	3013.4 ± 1206.0	<u>3538.1</u> ± 220.6	**3508.6** ± 272.7	3153.2 ± 520.8
Walker2d	PPO	5047.2 ± 591.6	**5156.3** ± 670.3	<u>5037.9</u> ± 1312.0	**5496.2** ± 125.9	<u>4858.3</u> ± 749.6
	PPO+SAM	<u>5139.1</u> ± 914.1	4591.1 ± 757.3	4792.2 ± 563.2	4394.5 ± 419.0	4689.4 ± 586.3
	PPO+mSAM	**5163.4** ± 563.2	<u>5106.1</u> ± 969.0	**5326.2** ± 1033.2	<u>5061.7</u> ± 628.9	**4932.4** ± 384.2
	PPO+NAG	4894.5 ± 890.5	4479.7 ± 970.4	4849.3 ± 477.1	4292.8 ± 271.5	4646.5 ± 359.5
	RARL	3522.7 ± 788.8	3499.4 ± 712.3	4464.3 ± 569.5	3838.6 ± 480.5	3563.0 ± 970.9
	RNAC	4707.4 ± 505.7	3797.8 ± 1583.4	4424.0 ± 342.6	4641.2 ± 212.4	4037.7 ± 750.2

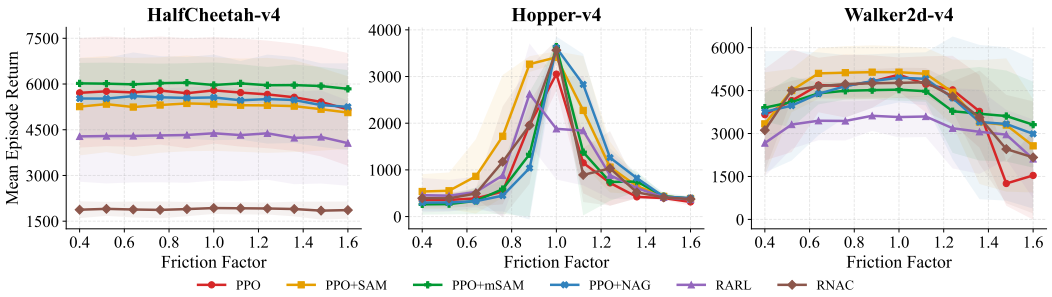


Figure 2: **Robustness evaluation under friction variations across different environments.** PPO+mSAM achieves the highest returns across the evaluated range on HalfCheetah-v4 and maintains strong performance at larger friction factors on Hopper-v4, where PPO and PPO+NAG deteriorate sharply. On Walker2d-v4, PPO+mSAM outperforms PPO at the largest friction factors, while PPO+SAM achieves the highest returns throughout the evaluated range.

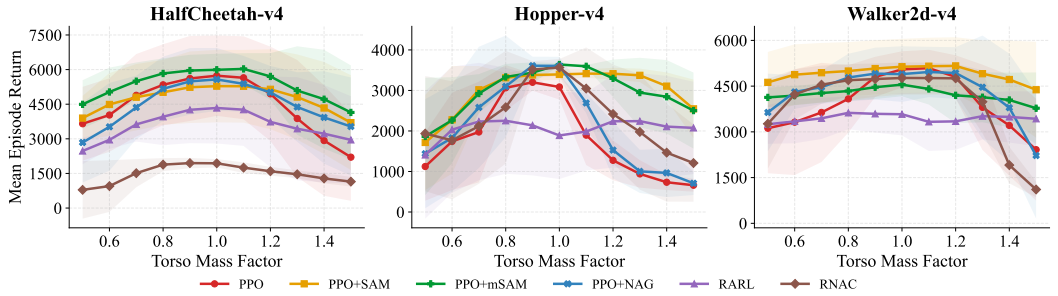


Figure 3: **Robustness evaluation under torso mass variations across different environments.** PPO+mSAM consistently achieves the highest returns on HalfCheetah-v4 and maintains strong performance under increased torso mass on Hopper-v4, where PPO and PPO+NAG degrade sharply. On Walker2d-v4, PPO+mSAM outperforms PPO at the largest mass factors, while PPO+SAM performs best throughout the evaluated range.

5.3 Rewards Surface Visualizations and Flatness Measurements

We visualize the reward landscape following (Sullivan et al., 2022). To quantify its flatness, we compute two metrics: the largest eigenvalue of the Hessian, $\lambda_{\max}$, and the low-pass filter (LPF) flatness measure (Bisla et al., 2022).

Flatness Metrics. We assess local geometry using Hessian eigenvalues and an LPF-inspired parameter sensitivity measure. Lower LPF values indicate lower relative sensitivity to parameter perturbations. Table 2 shows that SAM and mSAM generally yield flatter reward landscapes than PPO and NAG, as reflected by lower curvature and reduced sensitivity to parameter perturbations. SAM performs particularly well overall, while mSAM also achieves competitive flatness across environments.

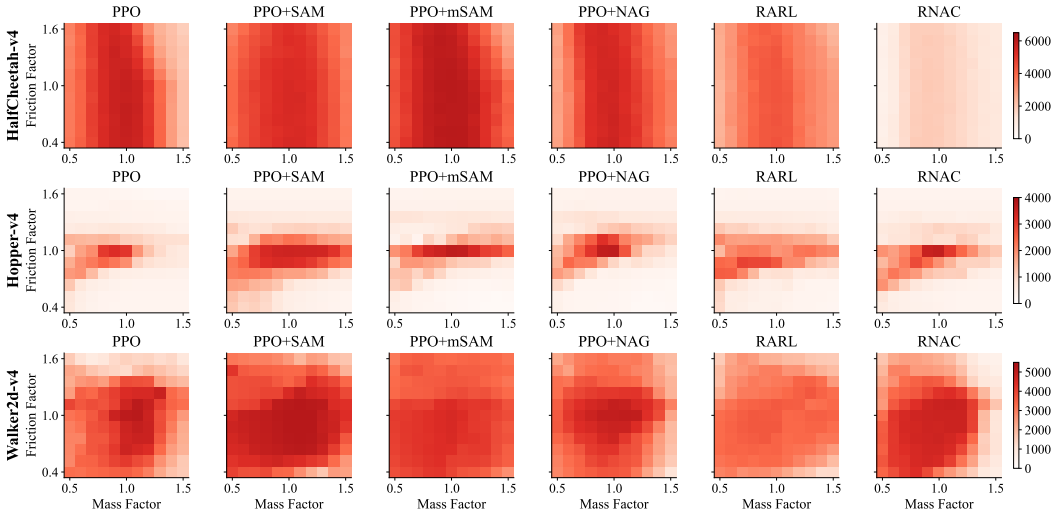


Figure 4: **Robustness evaluation under joint torso mass and friction variations across different environments.** PPO+mSAM maintains high returns over a broad region on HalfCheetah-v4. On Hopper-v4, PPO+mSAM and PPO+SAM tolerate a wider range of mass factors than PPO near nominal friction, but performance declines as friction deviates further from this setting. On Walker2d-v4, PPO+mSAM exhibits more uniform performance across the evaluated grid, while PPO+SAM achieves higher returns in much of the low-to-moderate friction region.

Reward Surface Visualization. We further visualize the reward landscape following the procedure of Sullivan et al. (2022). Figs. 16, 17 and 18 (see Appendix G.5) show that SAM generally exhibits broad, flat reward regions, while mSAM often yields wider regions of high return than PPO and NAG. Despite some variation across runs, these visualizations suggest that mSAM retains much of SAM's tendency to find locally flat solutions.

Table 2: **Flatness metrics of PPO, SAM, NAG, and mSAM.** Lower values indicate flatter reward landscapes. SAM and mSAM generally exhibit lower curvature and reduced sensitivity to parameter perturbations than PPO and NAG.

Environment	Method	LPF (Bisla et al., 2022)	$\lambda_{\max}$ ($\times 10^8$, Sullivan et al., 2022)
HalfCheetah-v4	PPO	0.086 ± 0.019	17.111 ± 11.967
	SAM	0.077 ± 0.043	1.418 ± 0.870
	NAG	0.095 ± 0.048	10.766 ± 6.761
	mSAM	0.086 ± 0.027	5.046 ± 5.324
Hopper-v4	PPO	0.229 ± 0.028	7.355 ± 4.660
	SAM	0.226 ± 0.116	0.112 ± 0.046
	NAG	0.199 ± 0.059	7.483 ± 5.830
	mSAM	0.149 ± 0.059	0.299 ± 0.222
Walker2d-v4	PPO	0.195 ± 0.054	2.479 ± 1.648
	SAM	0.144 ± 0.060	1.417 ± 0.566
	NAG	0.226 ± 0.138	3.747 ± 1.593
	mSAM	0.186 ± 0.076	0.448 ± 0.175

6 Conclusion

Seeking flat reward maxima offers a promising approach to robust reinforcement learning, but noisy and costly policy gradients make sharpness-aware optimization challenging in practice. We introduce mSAM for policy optimization, which uses accumulated momentum to guide adversarial parameter perturbations with only one policy-gradient evaluation per update. Reusing past gradients reduces reliance on a single noisy estimate and removes the extra gradient evaluation required by standard SAM. We further establish finite-time stationarity guarantees for SAM and mSAM, providing theoretical support for sharpness-aware policy learning. Experiments show that PPO+mSAM improves robustness over PPO under action and dynamics perturbations and learns effectively from noisy rewards on three MuJoCo benchmarks. It also requires less computation than PPO+SAM, making mSAM a simple and efficient approach to sharpness-aware policy learning.

Limitations & Future Work. Our results suggest several directions for further investigation. To what extent does local flatness in policy-parameter space translate into robustness to broader environmental changes? Our experiments provide encouraging evidence under action and dynamics perturbations, while a more detailed characterization could clarify how this connection depends on the induced changes in state visitation. Another question concerns the reliability of historical gradients during policy learning. Accumulated momentum reduces reliance on individual noisy estimates, although its relevance to the current objective may vary as the policy and value estimates evolve. Understanding this interaction could inform adaptive momentum and perturbation schedules that balance noise smoothing with responsiveness. Finally, our theoretical results characterize stationarity under smoothness and stochastic-gradient assumptions. Building on this analysis, explicitly accounting for approximate advantages, clipped objectives, and repeated data reuse would help clarify how the guarantees extend to practical actor–critic training and further explain the observed empirical behavior.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing, grammar and spelling checks, writing experimental scripts, and assisting with the analysis of experimental results. We did not use generative AI tools to generate the core research ideas, develop theoretical models, formulate mathematical claims, or develop or write mathematical proofs. All AI-assisted work was reviewed and cross-checked by the authors. We take full responsibility for the final content of this work, including text, claims, code, and analyses produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the theoretical assumptions and proofs in Appendices D and E, respectively. Experimental environments, perturbation protocols, model architectures, baseline implementations, and hyperparameters are described in Appendix F. Additional experiments and their corresponding settings are provided in Appendix G. We plan to release the source code to facilitate reproduction of the reported experimental results.

REFERENCES

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

Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In *International Conference on Machine Learning*, 2022.

Marlon Becker, Frederick Altrock, and Benjamin Risse. Momentum-SAM: Sharpness aware minimization without computational overhead. In *Advances in Neural Information Processing Systems*, 2025.

Devansh Bisla, Jing Wang, and Anna Choromanska. Low-pass filtering sgd for recovering flat optima in the deep learning optimization landscape. In *International Conference on Artificial Intelligence and Statistics*, 2022.

Severin Bochem, Eduardo Gonzalez-Sanchez, Yves Bicker, and Gabriele Fadini. Improving generalization of robot locomotion policies via sharpness-aware reinforcement learning. *arXiv preprint arXiv:2411.19732*, 2024.

Yen-Ju Chen, Nai-Chieh Huang, Ching-pei Lee, and Ping-Chun Hsieh. Accelerated policy gradient: on the convergence rates of the nesterov momentum for reinforcement learning. *arXiv preprint arXiv:2310.11897*, 2023.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, 2017.

Timothy Dozat. Incorporating nesterov momentum into adam. In *International Conference on Learning Representations, Workshop*, 2016.

Jiawei Du, Daquan Zhou, Jiashi Feng, Vincent Y. F. Tan, and Joey Tianyi Zhou. Sharpness-aware training for free. In *Advances in Neural Information Processing Systems*, 2022.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep rl: A case study on ppo and trpo. In *International Conference on Learning Representations*, 2019.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

Andrew Ilyas, Logan Engstrom, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. A closer look at deep policy gradients. In *International Conference on Learning Representations*, 2020.

Garud N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005.

Max Jaderberg, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, et al. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2019.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017.

Minyoung Kim, Da Li, Shell X Hu, and Timothy Hospedales. Fisher sam: Information geometry and sharpness aware minimisation. In *International Conference on Machine Learning*, 2022.

Ezgi Korkmaz. Nesterov momentum adversarial perturbations in the deep reinforcement learning domain. In *International Conference on Machine Learning*, 2020.

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. ASAM: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, 2021.

Hojoon Lee, Hanseul Cho, Hyunseung Kim, Daehoon Gwak, Joonkee Kim, Jaegul Choo, Se-Young Yun, and Chulhee Yun. PLASTIC: Improving input and label plasticity for sample efficient reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Hyun Kyu Lee and Sung Whan Yoon. Flat reward in policy parameter space implies robust reinforcement learning. In *International Conference on Learning Representations*, 2025.

Bingcong Li and Georgios B. Giannakis. Enhancing sharpness-aware optimization through variance suppression. In *Advances in Neural Information Processing Systems*, 2023.

Tao Li, Pan Zhou, Zhengbao He, Xinwen Cheng, and Xiaolin Huang. Friendly sharpness-aware minimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and scalable sharpness-aware minimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Kaustubh Mani, Yann Pequignot, Vincent Mai, and Liam Paull. SHAPO: Sharpness-aware policy optimization for safe exploration. In *International Conference on Learning Representations*, 2026.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvári, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, 2020.

Peng Mi, Li Shen, Tianhe Ren, Yiyi Zhou, Xiaoshuai Sun, Rongrong Ji, and Dacheng Tao. Make sharpness-aware minimization stronger: A sparsified perturbation approach. In *Advances in Neural Information Processing Systems*, 2022.

Goncalo Mordido, Pranshu Malviya, Aristide Baratin, and Sarath Chandar. Lookbehind-SAM: k steps back, 1 step forward. In *International Conference on Machine Learning*, 2024.

Jun Morimoto and Kenji Doya. Robust reinforcement learning. *Neural Computation*, 17(2):335–359, 2005.

Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

Dimitris Oikonomou and Nicolas Loizou. Sharpness-aware minimization: General analysis and improved rates. In *International Conference on Learning Representations*, 2025.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, 2017.

Md Masudur Rahman and Yexiang Xue. Robust policy optimization in deep reinforcement learning. *arXiv preprint arXiv:2212.07536*, 2022.

Shrinivas Ramasubramanian, Benjamin Freed, Alexandre Capone, and Jeff G. Schneider. Improving model-based reinforcement learning by converging to flatter minima. In *Advances in Neural Information Processing Systems*, 2025.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–489, 2016.

Ryan Sullivan, Justin K Terry, Benjamin Black, and John P Dickerson. Cliff diving: Exploring reward surfaces in reinforcement learning environments. *arXiv preprint arXiv:2205.07015*, 2022.

Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 1999.

Behrooz Tahmasebi, Ashkan Soleymani, Dara Bahri, Stefanie Jegelka, and Patrick Jaillet. A universal class of sharpness-aware minimization algorithms. *arXiv preprint arXiv:2406.03682*, 2024.

Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep reinforcement learning for robotics: A survey of real-world successes. *Annual Review of Control, Robotics, and Autonomous Systems*, 8(1):153–188, 2025.

Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning*, 2019.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

Mark Towers, Ariel Kwiatkowski, John Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Kallinteris Andreas, Markus Krimmel, Arjun Kg, Rodrigo Perez-Vicente, et al. Gymnasium: A standard interface for reinforcement learning environments. In *Advances in Neural Information Processing Systems*, 2025.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Da Wang, Yi Ma, Ting Guo, Lin Li, Wei Wei, and Jiye Liang. Q-SAM: Unlocking sharpness-aware minimization for generalization in offline reinforcement learning. In *International Conference on Machine Learning*, 2026.

Shengbo Wang, Chin Pang Ho, and Marek Petrik. Policy gradient in robust MDPs with global convergence guarantee. In *International Conference on Machine Learning*, 2023.

Yue Wang and Shaofeng Zou. Policy gradient method for robust reinforcement learning. In *International Conference on Machine Learning*, 2022.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992.

Le Xu and Jiayu Chen. Enhancing robustness of offline reinforcement learning under data corruption via sharpness-aware minimization. In *AAAI Conference on Artificial Intelligence*, 2026.

Runsheng Yu, Youzhi Zhang, and James T. Kwok. Improving sharpness-aware minimization by lookahead. In *International Conference on Machine Learning*, 2024.

Yihao Zhang, Hangzhou He, Jingyu Zhu, Huanran Chen, Yifei Wang, and Zeming Wei. On the duality between sharpness-aware minimization and adversarial training. *arXiv preprint arXiv:2402.15152*, 2024.

Yuan Zhang, Jianhong Wang, and Joschka Boedecker. Robust reinforcement learning in continuous control tasks with uncertainty set regularization. In *Conference on Robot Learning*, 2023.

Zhiyuan Zhang, Ruixuan Luo, Qi Su, and Xu Sun. Ga-sam: Gradient-strength based adaptive sharpness-aware minimization for improved generalization. *arXiv preprint arXiv:2210.06895*, 2022.

Mengnan Zhao, Lihe Zhang, Jingwen Ye, Huchuan Lu, Baocai Yin, and Xinchao Wang. Adversarial training: A survey. *arXiv preprint arXiv:2410.15042*, 2024.

Ruida Zhou, Tao Liu, Min Cheng, Dileep Kalathil, Panganamala R Kumar, and Chao Tian. Natural actor-critic for robust reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 2023.

APPENDICES

The appendices are structured as follows. In Appendix A, we provide the necessary background for our method. In Appendix B, we discuss the additional related work. In Appendix E, we present proofs of two of our main theoretical results. In Appendix F, we describe the experimental implementation, including environments, baselines, and hyperparameter settings. Finally, Appendix G includes additional experiments and the corresponding setup details.

LIST OF APPENDICES

A Background and Technicalities **15**
- A.1 Robust Reinforcement Learning . . . 15
- A.2 Sharpness-Aware Minimization . . . 16
- A.3 Relation between SAM and Adversarial Training . . . 17
- A.4 From Flat Policy Landscapes to Robustness. . . . 17

B Additional Related Work **18**
- B.1 Robust Reinforcement Learning . . . 18
- B.2 Sharpness-Aware Minimization and Its Variants . . . 19
- B.3 Flat Optimization Landscapes in Reinforcement Learning . . . 19

C Pseudocode **20**

D Assumptions and Additional Theoretical Results **22**
- D.1 Assumptions . . . 22
- D.2 Precise Case of Theorem 2 . . . 23
- D.3 On the Convergence of Policy Gradient with mSAM . . . 23
- D.4 Convergence Rate of Stationarity by SAM Policy Optimization . . . 23

E Proofs **25**
- E.1 Proof of Theorem 1 . . . 25
- E.2 Proof of Theorem 3 . . . 26
- E.3 Proof of Theorem 4 . . . 28
- E.4 Helpful Lemmas . . . 28

F Implementation of the Experiments **35**

G Additional Experiments and Setup Details **37**
- G.1 Experiments on Discrete Action Environments . . . 37
- G.2 Extension to Other Policy Learning Algorithms . . . 38
- G.3 Sensitivity Studies . . . 41
- G.4 Detailed Computational Efficiency Analysis . . . 42
- G.5 Rewards Surface Visualizations and Flatness Measurements . . . 43

A Background and Technicalities

Notations. Before proceeding, we summarize the main notation used throughout the paper in Table 3.

Table 3: Summary of notation.

Symbol	Description
$\boldsymbol{\theta}$	Policy parameters
$\pi_{\boldsymbol{\theta}}$	Policy parameterized by $\boldsymbol{\theta}$
μ	Initial-state distribution
γ	Discount factor
$V(\boldsymbol{\theta})$, $V^{\pi_{\boldsymbol{\theta}}}(\mu)$	Expected discounted return from μ
$V^{\pi_{\boldsymbol{\theta}}}(s)$	State-value function
$Q^{\pi_{\boldsymbol{\theta}}}(s,a)$	Action-value function
$A^{\pi_{\boldsymbol{\theta}}}(s,a)$	Advantage function
V^*	Minimum nominal objective value
$\Delta(\boldsymbol{\theta})$	Objective gap $V(\boldsymbol{\theta}) - V^*$
d_s^{π}	Discounted state-visitation distribution
$\mathcal{A}^*(s)$	Optimal action set at state s
$\mu_{\min}$	Minimum initial-state probability $\min_s \mu(s)$
ξ_t	Stochastic sample at iteration t
$\mathcal{B}_t$	Mini-batch at iteration t
$G(\boldsymbol{\theta}_t;\xi_t)$, $\boldsymbol{g}_t$	Stochastic policy-gradient estimator
$\boldsymbol{u}_t$	Normalized stochastic-gradient direction
ρ, ρ_t	SAM perturbation radius
$\boldsymbol{z}_t$	Adversarially perturbed parameters
$\boldsymbol{v}_t$	Momentum vector
β	Momentum coefficient
η, η_t	Step size
$\mathcal{F}_t$	Filtration of the optimization history
L	Smoothness constant of $V(\cdot)$
$K_{\eta,\rho,\alpha}$	Step-size perturbation coupling term
$N_{\eta,\rho,\alpha}$	Noise and perturbation-induced error term
$q_{\eta,\rho,\alpha}$	Value-gap growth factor

A.1 Robust Reinforcement Learning

Reinforcement learning aims to learn a policy that maximizes the expected cumulative reward through interactions with an environment. In many practical applications, however, the transition dynamics used during training may differ from those encountered after deployment because of modeling errors, parameter drift, or external disturbances. A policy optimized for a single nominal model can therefore be highly sensitive to such discrepancies. For an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, P, \gamma, d_0)$, the value of a policy π under transition model P is

$$V^{\pi}(s) = \mathbb{E}_{\pi, P\subset\mathcal{P}}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\,\middle|\, s_0 = s\right].$$

Robust RL addresses this problem through the robust Markov decision process framework, in which the true transition model is assumed to belong to an uncertainty set

$$\mathcal{P} = \otimes_{(s,a)\in\mathcal{S}\times\mathcal{A}}\mathcal{P}_{s,a}, \qquad \mathcal{P}_{s,a} = \left\{P_{s,a}\in\Delta(\mathcal{S}) : D(P_{s,a}, P^o_{s,a}) \le \epsilon_0\right\},$$

where P^o is the nominal model and ϵ_0 controls the degree of robustness. The objective is to find a policy that performs well under the worst admissible model,

$$\pi^\star \in \arg\max_{\pi}\inf_{P\in\mathcal{P}}\mathbb{E}_{s_0\sim d_0,\pi,P}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right].$$

The central difficulty of robust RL is to estimate worst-case performance over $\mathcal{P}$ using data generated only by the nominal model, while also handling distribution shift and function-approximation errors.

A.2 Sharpness-Aware Minimization

Recent studies have suggested that the sharpness of the loss landscapes, which describes how rapidly it changes in a neighborhood around the learned parameters, strongly correlates with generalization (Keskar et al., 2017; Jiang et al., 2019; Zhang et al., 2022). Building on this observation, recent works (Foret et al., 2021; Andriushchenko & Flammarion, 2022; Tahmasebi et al., 2024) have sought to minimize sharpness to improve generalization.

Sharpness. Sharpness (Keskar et al., 2017) measures how sensitively the loss changes under local parameter perturbations. Given a training loss $f(\mathbf{w})$ and a neighborhood radius $\rho > 0$, a worst-case measure of local sharpness is

$$S_\rho(\mathbf{w}) := \max_{\|\boldsymbol{\epsilon}\|_2 \le \rho} f(\mathbf{w}+\boldsymbol{\epsilon}) - f(\mathbf{w}).$$

A sharp solution admits small parameter perturbations that substantially increase the loss, whereas a flat solution exhibits a smaller worst-case increase within the same neighborhood. This distinction motivates seeking parameters that achieve both low training loss and low local sharpness, rather than minimizing the loss only at a single point.

Sharpness-Aware Minimization (SAM). Foret et al. (2021) implements this idea by jointly minimizing $f(\mathbf{w})$ and $S_\rho(\mathbf{w})$, yielding the following min–max objective:

$$\min_{\mathbf{w}} \Big\{ \max_{\|\boldsymbol{\epsilon}\|_2 \le \rho} f(\mathbf{w}+\boldsymbol{\epsilon}) \Big\}, \tag{8}$$

where $\lambda \ge 0$ is the regularization coefficient. By controlling the largest loss within the neighborhood, this formulation encourages parameters for which nearby configurations also maintain low loss.

The inner maximization in Eq. (8) identifies an adversarial parameter perturbation that maximizes the local loss. Since solving this problem exactly is generally expensive, Foret et al. (2021) use a first-order Taylor approximation of $f(\mathbf{w}+\boldsymbol{\epsilon})$ around $\boldsymbol{\epsilon} = \mathbf{0}$:

$$\begin{aligned}
\epsilon^\star(\mathbf{w}) &= \arg\max_{\|\boldsymbol{\epsilon}\|_2 \le \rho} f(\mathbf{w}+\boldsymbol{\epsilon}) \\
&\approx \arg\max_{\|\boldsymbol{\epsilon}\|_2 \le \rho} \{ f(\mathbf{w}) + \langle \nabla f(\mathbf{w}), \boldsymbol{\epsilon} \rangle \} \\
&= \rho \frac{\nabla f(\mathbf{w})}{\|\nabla f(\mathbf{w})\|_2}.
\end{aligned}$$

Hence, SAM implements this approximation through two gradient evaluations:

- The first gradient, evaluated at $\mathbf{w}$, is used to construct the approximate worst-case perturbation $\boldsymbol{\epsilon}$.
- The second gradient, evaluated at $\mathbf{w}+\boldsymbol{\epsilon}$, is used to update the original, unperturbed parameters.

Treating the constructed perturbation as fixed when computing the outer gradient, the update with gradient descent at iteration t is

$$\begin{aligned}
\boldsymbol{\epsilon}_t &\leftarrow \rho \nabla f(\mathbf{w}_t) / \|\nabla f(\mathbf{w}_t)\|_2, \\
\mathbf{w}_{t+1} &\leftarrow \mathbf{w}_t - \eta_t (\nabla f(\mathbf{w}_t + \boldsymbol{\epsilon}_t),
\end{aligned} \tag{9}$$

where η_t is the learning rate. In stochastic training, both gradients are typically evaluated using the same minibatch. More generally, the gradient at the perturbed parameters can be supplied to a base optimizer, making SAM compatible with different optimization algorithms. This procedure accounts for local parameter sensitivity, but requires an additional gradient evaluation per iteration relative to a standard update.

Despite its benefits, SAM has several limitations. First, each iteration requires two sequential gradient evaluations: one at the original parameters to construct the perturbation and another at

the perturbed parameters to determine the update. This dependency prevents the two evaluations from being performed in parallel, approximately doubling the gradient computation cost relative to base optimizers such as SGD and typically increasing training time substantially (Foret et al., 2021). Second, the perturbation is constructed from a stochastic minibatch gradient and is therefore sensitive to sampling noise in practice. When the gradient variance is high, the resulting direction may poorly approximate the local worst-case perturbation of the full training objective: a perturbation that increases the loss on one minibatch need not increase it on other samples (Li & Giannakis, 2023). Finally, stability under local parameter perturbations does not automatically imply robustness to environmental perturbations in reinforcement learning, and establishing this connection requires further analysis.

A.3 Relation between SAM and Adversarial Training

SAM and adversarial training (AT, Zhao et al., 2024) share a minimax optimization structure, with perturbations applied to model parameters and inputs, respectively. SAM seeks parameters with uniformly low loss in their neighborhoods to improve generalization (Foret et al., 2021), whereas AT minimizes the loss on adversarially perturbed inputs to improve input robustness (Madry et al., 2017). Their objectives can be written as

$$\textbf{SAM:} \quad \min_{\boldsymbol{\theta}} \max_{\|\boldsymbol{\epsilon}\|<\rho} \mathbb{E}_{(x,y)\sim\mathcal{D}} L(\boldsymbol{\theta}+\boldsymbol{\epsilon}; x, y), \tag{10}$$

$$\textbf{AT:} \quad \min_{\boldsymbol{\theta}} \max_{\|\delta\|<\varepsilon} \mathbb{E}_{(x,y)\sim\mathcal{D}} L(\boldsymbol{\theta}; x+\delta, y). \tag{11}$$

Despite this difference, both perturbation forms affect the model's feature representations. Zhang et al. (2024) connect the two methods by showing that both encourage greater reliance on robust features in a simplified classification model. This connection helps explain why SAM can improve adversarial robustness while maintaining clean accuracy in their experiments, although it does not establish equivalence with AT. AT targets input robustness more directly and can incur a trade-off with clean accuracy (Tsipras et al., 2018). This perspective provides a basis for examining parameter and input perturbations in reinforcement learning in the next section.

A.4 From Flat Policy Landscapes to Robustness.

The geometry of the optimization landscape provides a useful perspective on model robustness. In supervised learning, flat minima correspond to parameter regions in which small perturbations produce only minor changes in the loss, whereas sharp minima are highly sensitive to parameter variations. Sharpness-Aware Minimization (SAM) operationalizes this principle by minimizing the worst-case loss within a local parameter neighborhood:

$$\min_{\boldsymbol{\theta}} \max_{\|\boldsymbol{\epsilon}\|_2\leq\rho} L(\boldsymbol{\theta}+\boldsymbol{\epsilon}), \tag{12}$$

where $\boldsymbol{\epsilon}$ denotes a perturbation of the model parameters and ρ controls the size of the perturbation neighborhood. By optimizing the loss under the locally worst-case parameter perturbation, SAM encourages solutions that are less sensitive to parameter variations and often generalize better (Foret et al., 2021). Nevertheless, flatness should be interpreted carefully, since conventional sharpness measures can depend on the parameterization of the neural network rather than only on the underlying function represented by the model (Dinh et al., 2017).

A similar principle can be applied to reinforcement learning by seeking flat maxima of the expected return. A flat policy landscape indicates that the return remains high when the policy parameters are slightly perturbed, suggesting that the learned behavior does not rely on a narrowly tuned parameter configuration. To establish this connection, we express the PPO loss in terms of the negative expected discounted return:

$$L(\boldsymbol{\theta}) = -\mathbb{E}_{s\sim p,\, a\sim\pi_{\boldsymbol{\theta}}}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right]. \tag{13}$$

Here, the negative sign indicates that minimizing the loss corresponds to maximizing the expected cumulative return. Substituting Eq. (13) into the SAM objective in Eq. (12) gives

$$\min_{\boldsymbol{\theta}} \max_{\|\boldsymbol{\epsilon}\|_2\leq\rho} \left\{\mathbb{E}_{s\sim p,\, a\sim\pi_{\boldsymbol{\theta}+\boldsymbol{\epsilon}}}\left[\sum_{t=0}^{\infty}-\gamma^t r(s_t, a_t)\right]\right\}. \tag{14}$$

The inner maximization identifies the parameter perturbation within the radius ρ that produces the largest reduction in return, whereas the outer minimization searches for policy parameters that perform well even under this worst-case perturbation. Therefore, Eq. (14) encourages the policy to converge to a flat reward maximum in parameter space. Such a policy is locally insensitive to parameter perturbations, which can also reduce the sensitivity of its action distribution. Recent empirical results support this connection, showing that policies located in flatter reward regions can be more resilient to action noise and, in some cases, to variations in transition dynamics and reward functions (Lee & Yoon, 2025).

This interpretation establishes a connection with the Action Robust MDP framework, which explicitly considers deviations between the action selected by the policy and the action executed in the environment (Tessler et al., 2019). Its objective is to find a policy that maximizes the minimum expected return under all allowable action perturbations:

$$\min_{\pi} \max_{\|\delta_t\|_2 \le \rho} \mathbb{E}_{s\sim p,\, a\sim\pi} \left[\sum_{t=0}^{\infty} -\gamma^t r(s_t, a_t + \delta_t) \right], \tag{15}$$

where δ_t denotes the perturbation applied to the action at time t, and $\beta \ge 0$ bounds the magnitude of the admissible action perturbation. Eq. (14) and Eq. (15) share the same min-max robust-optimization principle. Eq. (14) considers the worst-case perturbation $\boldsymbol{\epsilon}$ in policy-parameter space, whereas Eq. (15) considers the worst-case perturbation δ_t in action space. Because perturbing the policy parameters changes the actions generated by the policy, robustness within a local parameter neighborhood can be transferred to robustness against the corresponding action perturbations. In this sense, the parameter radius ρ and the action radius β play analogous, although not generally identical, roles in the two objectives.

Consequently, SAM-based flat-policy optimization can be interpreted as an indirect, model-free approach to action robustness: instead of explicitly injecting adversarial action perturbations or training a separate adversary, it searches for policy parameters whose expected return remains stable under worst-case local parameter perturbations. This linkage is nevertheless local and depends on how changes in the policy parameters affect the resulting action distribution. Moreover, the connection to transition and reward robustness is primarily intuitive and empirical, while the most direct theoretical connection is between parameter-space flatness and action robustness.

B Additional Related Work

B.1 Robust Reinforcement Learning

Robust reinforcement learning aims to learn policies whose performance remains reliable under model misspecification, environmental disturbances, or adversarial perturbations. Classical robust Markov decision processes formalize this objective through a max–min problem in which a policy is evaluated against the worst transition kernel or reward function in a prespecified uncertainty set. Under suitable rectangularity assumptions, the resulting robust control problem admits dynamic-programming characterizations analogous to those of standard MDPs (Iyengar, 2005; Nilim & El Ghaoui, 2005). More recent work has extended this perspective to high-dimensional and model-free settings. For example, robust adversarial reinforcement learning explicitly trains an adversary that perturbs the environment (Pinto et al., 2017), while action-robust RL models uncertainty directly in the executed actions (Tessler et al., 2019). Robust policy-gradient methods instead optimize the worst-case return over transition uncertainty and provide asymptotic or finite-time global convergence guarantees under tabular or direct policy parameterizations (Wang & Zou, 2022; Wang et al., 2023).

These approaches primarily define robustness in the environment space: uncertainty is imposed on transition kernels, rewards, disturbances, or actions, and the learning algorithm explicitly optimizes over the corresponding uncertainty set. Although this formulation provides a direct notion of worst-case performance, solving or approximating the inner adversarial problem can be computationally expensive. In contrast, our work investigates robustness through the local geometry of the policy-value function in parameter space. This provides a complementary perspective in which robustness is encouraged by reducing the sensitivity of the learned policy to perturbations of its parameters.

B.2 Sharpness-Aware Minimization and Its Variants

Several SAM variants improve the quality or efficiency of the inner perturbation. Lookahead-SAM uses extrapolation to mitigate unstable minimax dynamics (Yu et al., 2024), and Lookbehind-SAM combines multiple ascent steps with interpolation to improve the loss-sharpness trade-off (Mordido et al., 2024). Most directly related, Momentum-SAM constructs perturbations from the optimizer's momentum and avoids the extra gradient pass used by standard SAM in supervised learning (Becker et al., 2025). Approaches in (Kim et al., 2022) and (Kwon et al., 2021) modify SAM to address coordinate-wise ill-scaling. Computationally efficient SAM variants reduce its additional overhead by reusing or sparsifying gradients (Liu et al., 2022; Mi et al., 2022), introducing stochastic perturbations (Du et al., 2022). Our Momentum SAM can be viewed as a resource-efficient SAM variant that reuses the optimizer's accumulated momentum to approximate the ascent direction, thereby avoiding the additional gradient computation required solely for perturbation construction.

Most of this literature studies supervised objectives or generic stochastic optimization problems. Its conclusions cannot be transferred directly to policy gradient, where the sampling distribution depends on the current policy and where the value objective is generally nonconcave. Analyzing SAM in reinforcement learning consequently requires explicit treatment of trajectory sampling, policy-dependent gradient noise, and the difference between the nominal and perturbed policies.

B.3 Flat Optimization Landscapes in Reinforcement Learning

A growing body of evidence indicates that the geometry of the optimization landscape is also important in reinforcement learning. Visualizations of policy reward surfaces reveal that many RL environments contain sharp cliffs, across which small parameter changes can lead to large performance degradation; trust-region-style methods such as PPO appear less likely to cross these cliffs than unconstrained policy-gradient updates (Sullivan et al., 2022). From a continual-learning perspective, changes in curvature and loss of plasticity have also been identified as important failure modes of deep RL agents. More directly, recent work has connected flat reward landscapes in policy parameter space to robust control. Lee & Yoon (2025) show that a reward function that is locally insensitive to policy-parameter perturbations induces robustness to action perturbations and can consequently improve robustness to changes in transition dynamics and reward functions. In model-based RL, encouraging flat minima in the world-model loss has been shown to reduce value-prediction and policy-performance errors. PAC-Bayesian bounds provide a formal connection between model sharpness and downstream control performance (Ramasubramanian et al., 2025). Sharpness-aware policy perturbations have also been used as a proxy for actor-side epistemic uncertainty, resulting in safer and more conservative exploration in poorly observed regions (Mani et al., 2026).

Collectively, these studies suggest that flatness is not merely a generalization property imported from supervised learning. In RL, a flat reward or model-loss landscape can represent reduced sensitivity to policy, action, model, and environmental perturbations, and can therefore serve as a mechanism for robustness. Nevertheless, existing studies either assume flatness and derive its robustness consequences, or provide empirical evidence that sharpness-aware training improves robustness. They do not generally characterize how a stochastic SAM policy-gradient method reaches such a region, nor how its optimization error depends on the perturbation radius and policy-gradient noise.

Positioning of Our Work. Our work connects these three lines of research by using SAM as an optimization and analytical mechanism for robust policy learning. Rather than directly solving a robust MDP through an explicit inner optimization over transition or reward uncertainty, we perturb the policy parameters and optimize the policy gradient evaluated at the locally adverse parameter. This formulation allows us to quantify the trade-off between nominal optimization accuracy and local parameter-space robustness. In particular, our convergence analysis separates the transient optimization error from the residual terms caused by the SAM radius and stochastic policy-gradient noise. When combined with a structural result that transfers flatness in policy-parameter space to robustness against action or environment perturbations, the analysis provides a principled explanation for why SAM-induced flatness can benefit robust reinforcement learning.

C PSEUDOCODE

Policy Optimization with SAM. To incorporate SAM into policy learning, we minimize a policy loss using gradients evaluated at locally perturbed parameters. At each iteration, SAM first computes the batch gradient $\boldsymbol{g}_k = G_k(\boldsymbol{\theta}_k)$ using a trajectory batch $\mathcal{B}_k$. It then constructs the locally adverse parameters $\boldsymbol{z}_k = \boldsymbol{\theta}_k + \rho \boldsymbol{u}_k$, where $\boldsymbol{u}_k$ is the normalized gradient direction, with $\boldsymbol{u}_k = \mathbf{0}$ when $\boldsymbol{g}_k = \mathbf{0}$. This perturbation approximates the inner maximization of the policy loss within a neighborhood of radius ρ. Using the same batch, SAM computes the gradient at $\boldsymbol{z}_k$ and applies the resulting update to the nominal parameters $\boldsymbol{\theta}_k$. Each iteration therefore requires two gradient evaluations: one to construct the perturbation and one to update the policy. The complete procedure is summarized in Algorithm 1.

Algorithm 1 Policy Optimization + SAM

Input: Training data $\mathcal{S}$, initial policy parameters $\boldsymbol{\theta}_0$, learning rate η, perturbation radius ρ, batch size B, and number of iterations T.

1: **for** $k = 0, 1, \ldots, K-1$ **do**
2: Sample a trajectory batch $\mathcal{B}_k = \{\tau_i\}_{i=1}^{B} \subseteq \mathcal{S}$.
3: Compute the gradient estimate of the policy loss:

$$\boldsymbol{g}_k := G_k(\boldsymbol{\theta}_k) = \frac{1}{B} \sum_{i=1}^{B} \hat{g}(\boldsymbol{\theta}_k, \tau_i).$$

4: Compute the normalized perturbation direction:

$$\boldsymbol{u}_k = \boldsymbol{g}_k / \|\boldsymbol{g}_k\|, \quad \boldsymbol{g}_k \neq \mathbf{0}.$$

5: Compute the perturbed parameters: $\boldsymbol{z}_k = \boldsymbol{\theta}_k + \rho \boldsymbol{u}_k$.
6: Compute the SAM gradient using the same batch $\mathcal{B}_k$: $\boldsymbol{g}_k^{\mathrm{SAM}} = G_k(\boldsymbol{z}_k) = \frac{1}{B} \sum_{i=1}^{B} \hat{g}(\boldsymbol{z}_k, \tau_i)$.
7: Update the policy parameters: $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta \boldsymbol{g}_k^{\mathrm{SAM}}$.
8: **end for**

PPO with mSAM. To illustrate how mSAM integrates with policy learning, we apply it to proximal policy optimization (PPO). Each rollout iteration proceeds as follows:

1. **Collect Trajectories:** Set $\theta_{\text{old}} = \theta$ and collect $\mathcal{B} = \{\tau_i\}_{i=1}^{N}$ under $\pi_{\theta_{\text{old}}}$.
2. **Compute Advantages:** Compute advantage estimates $\widehat{A}_j$ and value targets $\widehat{R}_j$ using the trajectories and value function V_ϕ.
3. **Construct Policy Loss:** Form the negative PPO clipped objective

$$L_{\mathcal{B}_k}(\theta) = -\frac{1}{|\mathcal{B}_k|} \sum_{(s,a,r,s') \in \mathcal{B}_k} \min\Big(\frac{\pi_\theta(a \mid s)}{\pi_{\theta_k}(a \mid s)} A^{\pi_{\theta_k}}(s,a),\ g\big(\delta, A^{\pi_{\theta_k}}(s,a)\big) \Big);$$

   where $\delta > 0$ is the PPO clipping threshold and

$$g(\delta, A) = \begin{cases} (1+\delta)A, & A \geq 0, \\ (1-\delta)A, & A < 0. \end{cases}$$

4. **Construct Perturbation and Compute Gradient:** Set $\epsilon = -\rho \boldsymbol{v}/\|\boldsymbol{v}\|$, and evaluate $\nabla_\theta L_{\mathcal{B}}(\theta)$ at $\theta + \epsilon$.
5. **Update Policy:** Restore the nominal parameters θ and use the computed gradient to update the momentum and policy.
6. **Fit Value Function:** Update ϕ by minimizing the mean-squared error between $V_\phi(s_t)$ and the fixed targets $\widehat{R}_t$.

Algorithm 2 PPO+mSAM

Input: initial policy π_{θ_0}, initial value function V_{ϕ_0}, learning rate η, momentum coefficient β.

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Collect set of trajectories $\mathcal{B}_k = \{(s, a, r, s')\}$ by running policy π_{θ_k} in the environment;
3: Compute advantage estimates $\hat{A}_k$;
4: Compute the policy PPO objective:

$$L_{\mathcal{B}_k}(\theta) = -\frac{1}{|\mathcal{B}_k|} \sum_{(s,a,r,s') \in \mathcal{B}_k} \min\left(\frac{\pi_\theta\,(a \mid s)}{\pi_{\theta_k}\,(a \mid s)} A^{\pi_{\theta_k}}\,(s,a)\,,\ g\,(\epsilon, A^{\pi_{\theta_k}}\,(s,a)) \right);$$

5: Update the policy π_{θ_k} using mSAM:

$$\boldsymbol{v}_{k+1} = \beta \boldsymbol{v}_k + \nabla L_{\mathcal{B}_k}\big(\theta_k - \rho \boldsymbol{v}_k/\|\boldsymbol{v}_k\|\big),$$
$$\theta_{k+1} = \theta_k - \eta \boldsymbol{v}_{k+1}.$$

6: Update ϕ via gradient descent on $\mathcal{L}_\phi = \frac{1}{|\mathcal{B}_k|T} \sum_{\tau \in \mathcal{B}_k} \sum_{t=0}^{T} \left(V_\phi\,(s_t) - \hat{R}_t\right)^2$.
7: **end for**

D Assumptions and Additional Theoretical Results

D.1 Assumptions

Assumption 1. *Let $\mathcal{F}_t := \sigma(\boldsymbol{\theta}_0, \boldsymbol{v}_0, \xi_0, \ldots, \xi_{t-1})$. If ξ_t and $\mathcal{F}_t$ are conditional independent, and for any $\mathcal{F}_t$-measurable vector $\boldsymbol{\theta}_t$, then*

$$\mathbb{E}\big[\nabla V(\boldsymbol{\theta}; \xi_t) \mid \mathcal{F}_t\big] = \nabla V(\boldsymbol{\theta}). \tag{16}$$

Assumption 2. *There exist constants $c_1, c_2 > 0$, such that for any $\mathcal{F}_t$-measurable $\boldsymbol{\theta}$, we have*

$$\mathbb{E}\left[\|\nabla V(\boldsymbol{\theta}; \xi_t)\|^2 \mid \mathcal{F}_t\right] \le c_1 \|\nabla V(\boldsymbol{\theta})\|^2 + c_2. \tag{17}$$

Remark. *In our setting, Assumption 1 is an unbiased gradient condition, which is mild and holds generally. Assumption 2 is about the second-order norm of the gradients, also known as a variance condition. Under our settings, it is reasonable to assume that $c_1 = 1, c_2 = \frac{\sigma^2}{B}$ if necessary, where σ^2 is the variance and B is the batch size.*

Assumption 3 (Conditions for optimal-action mass and gradient domination)**.** *The following conditions hold.*

1. *The discounted MDP has finite state and action spaces $\mathcal{S}$ and $\mathcal{A}$, with discount factor $\gamma \in (0, 1)$. The policy is parameterized by tabular softmax, $\pi_{\boldsymbol{\theta}}(a \mid s) = \frac{\exp(\theta(s,a))}{\sum_{b \in \mathcal{A}} \exp(\theta(s,b))}$. The distribution μ used in the optimization objective has full support*

$$\mu_{\min} := \min_{s \in \mathcal{S}} \mu(s) > 0.$$

2. *For each state s, define*

$$\mathcal{A}^*(s) := \{a \in \mathcal{A} : Q^*(s,a) = V^*(s)\}, \qquad p_t^*(s) := \sum_{a \in \mathcal{A}^*(s)} \pi_{\boldsymbol{\theta}_t}(a \mid s).$$

Let $\mathcal{S}_{\text{sub}} := \{s \in \mathcal{S} : \mathcal{A}^*(s) \neq \mathcal{A}\}$. For every $s \in \mathcal{S}_{\text{sub}}$, the statewise optimality gap is strictly positive:

$$\Delta_s := \min_{a \notin \mathcal{A}^*(s)} \big[Q^*(s,a) - V^*(s)\big] > 0.$$

Define $\Gamma_\mu := \min_{s \in \mathcal{S}_{\text{sub}}} \mu(s)\Delta_s > 0$. If $\mathcal{S}_{\text{sub}} = \varnothing$, every action is optimal at every state, and the optimal-action-mass conclusions are immediate.

3. *The objective $V(\cdot)$ is L-smooth. Moreover,*

$$\beta \in [0, 1), \qquad \eta_t > 0, \qquad \rho_t \ge 0, \qquad \sum_{t=0}^{\infty} \eta_t \rho_t^2 < \infty.$$

There exists a constant $c_\eta > 0$ such that, for every $t \ge T \ge 1$, the iterates satisfy

$$\delta_t + \frac{\beta \eta_{t-1}}{2} \|\boldsymbol{v}_t\|^2 \le \delta_T + \frac{\beta \eta_{T-1}}{2} \|\boldsymbol{v}_t\|^2 + \frac{L^2}{2c_\eta} \sum_{k=t-1}^{T} \eta_k \rho_k^2. \tag{18}$$

Remark. *The assumptions underlying the preceding global-rate analysis serve different purposes and should be interpreted separately. First, the finite tabular MDP, bounded rewards, and the softmax parameterization provide a controlled setting in which the policy objective is globally smooth and its gradient is uniformly bounded. The full-support condition $\mu(s) > 0$ ensures that every state is with less distribution-mismatch. This condition may nevertheless be restrictive in large-scale reinforcement learning, where adequate state coverage is itself a nontrivial exploration problem.*

A time-independent convergence rate requires a uniform trajectory-dependent lower bound

$$\underline{p} := \inf_{t \ge 0} p(\theta_t) > 0, \quad p(\theta_t) := \min_s \pi_{\theta_t}(a^*(s) \mid s).$$

The uniqueness of the optimal action at every state implies that global value convergence forces $p(\theta_t) \to 1$. Hence, if global value convergence or the probability floor $\underline{p} > 0$ is imposed as an assumption, the resulting $O(1/t)$ rate is a conditional global-optimality result.

D.2 Precise Case of Theorem 2

Theorem 3 (Finite-time stationarity of mSAM with variable step sizes). *Under Assumptions 1 and 2 and Algorithm 2, if for every $t \geq 0$, $\eta_t \leq \frac{(1-\beta)^3}{L\left(\beta^2+2c_1(1-\beta)^2\right)}$, Define $\alpha_t := \sum_{k=0}^{\infty} \beta^k \eta_{t+k}$, $c_{\beta,t} := \beta\alpha_t$. Then, for every integer $T \geq 1$, we have*

$$\min_{0\leq t<T} \mathbb{E}\|\nabla V(\theta_t)\|^2 \leq \frac{2\Delta(\theta_0)}{\sum_{t=0}^{T-1}\alpha_t} + Lc_2\left(1+\frac{\beta^2}{c_1(1-\beta)^2}\right)\frac{\sum_{t=0}^{T-1}\alpha_t^2}{\sum_{t=0}^{T-1}\alpha_t} + L^2\frac{\sum_{t=0}^{T-1}\alpha_t\rho_t^2}{\sum_{t=0}^{T-1}\alpha_t}. \tag{19}$$

In terms of the original step sizes, the following bound also holds,

$$\min_{0\leq t<T} \mathbb{E}\|\nabla V(\theta_t)\|^2 \leq \frac{2\Delta(\theta_0)}{\sum_{t=0}^{T-1}\eta_t} + \frac{Lc_2}{(1-\beta)^2}\left(1+\frac{\beta^2}{c_1(1-\beta)^2}\right)\frac{\sum_{t=0}^{T-1}\eta_t^2}{\sum_{t=0}^{T-1}\eta_t} + \frac{L^2}{1-\beta}\frac{\sum_{t=0}^{T-1}\eta_t\rho_t^2}{\sum_{t=0}^{T-1}\eta_t}. \tag{20}$$

In particular, when $c_1 = 1$ and $c_2 = \sigma^2/B$, we have

$$\min_{0\leq t<T} \mathbb{E}\|\nabla V(\theta_t)\|^2 \leq \frac{2\Delta(\theta_0)}{\sum_{t=0}^{T-1}\eta_t} + \frac{L\sigma^2}{B(1-\beta)^2}\left(1+\frac{\beta^2}{(1-\beta)^2}\right)\frac{\sum_{t=0}^{T-1}\eta_t^2}{\sum_{t=0}^{T-1}\eta_t} + \frac{L^2}{1-\beta}\frac{\sum_{t=0}^{T-1}\eta_t\rho_t^2}{\sum_{t=0}^{T-1}\eta_t}. \tag{21}$$

D.3 On the Convergence of Policy Gradient with mSAM

Theorem 4 (Global Convergence Analysis of Policy Gradient with mSAM). *Under Algorithm 2 with true gradient $\nabla V(\boldsymbol{\theta}_t)$, suppose that $\mu_{\min} > 0$, and the variable step sizes and perturbation radii satisfy Eq. (85), and that the conditions in Eq. (90) holds at some finite time T. Then we have*

$$V^{\pi_{\boldsymbol{\theta}_t}}(\mu) \longrightarrow V^*(\mu), \tag{22}$$

for every initial-state distribution μ.

Remark. *The existing analyses in Theorem 4 do not justify an unconditional global-convergence claim. This global convergence result requires more conditions such as a full-support initial-state distribution, strictly positive optimality gaps, objective smoothness, summability conditions on the step sizes and perturbation radii, and eventual entry into an energy region where the optimal-action probabilities remain uniformly bounded away from zero. Condition Assumption 3 is a sufficient one-time condition, not a modification of the update rule. It says that at some iteration the current value gap, momentum energy, and remaining perturbation budget are jointly smaller than the smallest distribution-weighted optimality gap Γ_μ. For vanilla softmax policy gradient, Agarwal et al. (2021) proves global value convergence from arbitrary finite initialization by exploiting a special componentwise monotonicity of the vanilla update. The probability lower bound in Mei et al. (2020) is subsequently obtained from that value convergence and positivity over a finite prefix. Momentum and the perturbed gradient generally destroy the required monotonicity. Therefore, without either condition Assumption 3 or a new lemma showing that the momentum-SAM dynamics enters this region from every initialization, the preceding argument does not establish unconditional global convergence.*

D.4 Convergence Rate of Stationarity by SAM Policy Optimization

Corollary 1. *Fix constants $s, r > 0$, if set $\eta = \frac{s}{\sqrt{T}}$, $\rho = \frac{r}{\sqrt{T}}$, and assume T is large enough that the step size condition holds. Then*

$$\min_{0\leq t<T} \mathbb{E}\left\|\nabla V(\boldsymbol{\theta}_t)\right\|^2 \leq \frac{\Delta(\boldsymbol{\theta}_0)}{s\sqrt{T}} + \frac{2L}{\sqrt{T}}\left[c_2 s + r\left(1+\frac{2srL^2}{T}\right)\left(\alpha^2 + C(1-\alpha)^2\right)\right]. \tag{23}$$

This implies $\min_{0\le t<T} \mathbb{E}\,\|\nabla V(\boldsymbol{\theta}_t)\|^2 = O(T^{-1/2})$.

In particular, obtaining an iterate satisfying $\mathbb{E}\,\|\nabla V(\boldsymbol{\theta}_t)\| \le \varepsilon$ requires $T = O(\varepsilon^{-4})$ iterations, up to problem-dependent constants.

Proof of Corollary 1. Substituting the expression of η, rh into $N_{\eta,\rho,\alpha}$ and using Jensen's inequality gives the results. Moreover, since $\mathbb{E}\,\|\nabla V(\boldsymbol{\theta}_t)\| \le \sqrt{\mathbb{E}\,\|\nabla V(\boldsymbol{\theta}_t)\|^2}$, we have the right-hand side of order $T^{-1/2}$ for the squared norm becomes order $T^{-1/4}$ for the expected norm, proving the stated $O(\varepsilon^{-4})$ iteration complexity. □

E PROOFS

E.1 PROOF OF THEOREM 1

Proof of Theorem 1. Firstly, we rewrite the algorithm as the following form,

$$\begin{aligned} \boldsymbol{g}_t &:= G(\boldsymbol{\theta}_t;\xi_t), \quad \boldsymbol{u}_t := \begin{cases} \boldsymbol{g}_t/\left\|\boldsymbol{g}_t\right\|, & \boldsymbol{g}_t \neq \mathbf{0}, \\ \mathbf{0}, & \boldsymbol{g}_t = \mathbf{0}, \end{cases} \\ \boldsymbol{z}_t &:= \boldsymbol{\theta}_t - \rho\big((1-\alpha)\boldsymbol{g}_t + \alpha \boldsymbol{u}_t\big), \\ \boldsymbol{g}_t^{\mathrm{SAM}} &:= G(\boldsymbol{z}_t;\xi_t), \quad \boldsymbol{\theta}_{t+1} := \boldsymbol{\theta}_t - \eta \boldsymbol{g}_t^{\mathrm{SAM}}, \end{aligned} \tag{24}$$

and give some notations for compactness

$$\begin{aligned} K_{\eta,\rho,\alpha} &:= \eta + \rho(1-\alpha)^2\big(1+2\eta\rho L^2\big), \\ N_{\eta,\rho,\alpha} &:= c_2\eta + \rho\big(1+2\eta\rho L^2\big)\big(\alpha^2 + c_2(1-\alpha)^2\big). \end{aligned} \tag{25}$$

The choice $\alpha = 1$ is normalized SAM, whereas $\alpha = 0$ is the unnormalized variant. Let $\mathcal{F}_t$ denote the sigma-field generated by all randomness before sampling ξ_t, and write $\mathbb{E}[\cdot] := \mathbb{E}[\cdot \mid \mathcal{F}_t]$. The L-smoothness of $V(\cdot)$ and the descent update Eq. (24) imply

$$V(\boldsymbol{\theta}_{t+1}) \leq V(\boldsymbol{\theta}_t) - \eta\langle \nabla V(\boldsymbol{\theta}_t), \boldsymbol{g}_t^{\mathrm{SAM}}\rangle + \frac{L\eta^2}{2}\left\|\boldsymbol{g}_t^{\mathrm{SAM}}\right\|^2. \tag{26}$$

By taking conditional expectation, we obtain

$$\mathbb{E}[\Delta(\boldsymbol{\theta}_{t+1})] - \Delta(\boldsymbol{\theta}_t) \leq -\eta\mathbb{E}[\langle \boldsymbol{g}_t^{\mathrm{SAM}}, \nabla V(\boldsymbol{\theta}_t)\rangle] + \frac{L\eta^2}{2}\mathbb{E}[\left\|\boldsymbol{g}_t^{\mathrm{SAM}}\right\|^2]. \tag{27}$$

Substituting Lemma 6 and Lemma 7 into Eq. (27), and collecting the terms multiplying $\mathbb{E}[\|\boldsymbol{g}_t\|^2]$ gives

$$\begin{aligned} \mathbb{E}[\Delta(\boldsymbol{\theta}_{t+1})] - \Delta(\boldsymbol{\theta}_t) \leq & -\eta\left(1 - \frac{L\rho}{2}\right)\left\|\nabla V(\boldsymbol{\theta}_t)\right\|^2 \\ & + \eta L K_{\eta,\rho,\alpha}\mathbb{E}_t\left\|\boldsymbol{g}_t\right\|^2 + \eta L\rho\big(1+2\eta\rho L^2\big)\alpha^2. \end{aligned} \tag{28}$$

Using Assumption 1 in Eq. (28) yields

$$\begin{aligned} \mathbb{E}_t[\Delta(\boldsymbol{\theta}_{t+1})] - \Delta(\boldsymbol{\theta}_t) \leq & -\eta\left(1 - \frac{L\rho}{2} - c_1 L K_{\eta,\rho,\alpha}\right)\left\|\nabla V(\boldsymbol{\theta}_t)\right\|^2 \\ & + \eta L N_{\eta,\rho,\alpha}. \end{aligned} \tag{29}$$

We next verify the descent coefficient. By the step size condition of the theorem, we have

$$\frac{L\rho}{2} \leq \frac{1}{8}, \quad c_1 L\rho(1-\alpha)^2 \leq \frac{1}{8}, \quad c_1 L\eta \leq \frac{1}{8},$$

and $2c_1\eta L^3\rho^2(1-\alpha)^2 \leq \frac{(1-\alpha)^2}{64} \leq \frac{1}{64}$. Consequently, we get

$$1 - \frac{L\rho}{2} - c_1 L K_{\eta,\rho,\alpha} \geq \frac{1}{2}. \tag{30}$$

Taking total expectation in Eq. (29), and using Eq. (30) gives

$$\frac{\eta}{2}\mathbb{E}\left\|\nabla V(\boldsymbol{\theta}_t)\right\|^2 \leq \Delta_t - \Delta_{t+1} + \eta L N_{\eta,\rho,\alpha}. \tag{31}$$

We next telescope the slightly expansive recursion on Eq. (31) , which yields

$$\begin{aligned} \frac{\eta}{2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla V(\boldsymbol{\theta}_t)\right\|^2 &\leq \Delta_0 - \Delta_T + \eta L N_{\eta,\rho,\alpha}\sum_{t=0}^{T-1} \\ &\leq \Delta_0 + \eta L N_{\eta,\rho,\alpha}\sum_{t=0}^{T-1}. \end{aligned}$$

Dividing by the sum of weights and then upper-bounding its reciprocal by $1/T$ proves

$$\min_{0\leq t<T}\mathbb{E}\left\|\nabla V(\boldsymbol{\theta}_t)\right\|^2 \leq \frac{\Delta_0}{T\eta} + 2L N_{\eta,\rho,\alpha},$$

which proves Eq. (6). □

E.2 Proof of Theorem 3

Proof of Theorem 3. Define the auxiliary sequence $\boldsymbol{y}_t := \boldsymbol{\theta}_t - c_{\beta,t}\boldsymbol{v}_t$, and the time-dependent Lyapunov function

$$\Phi_t := V(\boldsymbol{y}_t) - V^* + \frac{Lc_{\beta,t}^2}{2c_1(1-\beta)}\|\boldsymbol{v}_t\|^2. \tag{32}$$

By Lemma 5, we have $\alpha_{t+1} \le \alpha_t, 0 \le c_{\beta,t+1} \le c_{\beta,t}$ This implies

$$\begin{aligned}\mathbb{E}[\Phi_{t+1} \mid \mathcal{F}_t] &= \mathbb{E}[V(\boldsymbol{y}_{t+1}) \mid \mathcal{F}_t] - V^* \\ &\quad + \frac{Lc_{\beta,t+1}^2}{2c_1(1-\beta)}\mathbb{E}[\|\boldsymbol{v}_{t+1}\|^2 \mid \mathcal{F}_t] \\ &\le \mathbb{E}[V(\boldsymbol{y}_{t+1}) \mid \mathcal{F}_t] - V^* \\ &\quad + \frac{Lc_{\beta,t}^2}{2c_1(1-\beta)}\mathbb{E}[\|\boldsymbol{v}_{t+1}\|^2 \mid \mathcal{F}_t].\end{aligned} \tag{33}$$

This inequality explicitly accounts for the change in the Lyapunov coefficient between consecutive iterations. Applying Lemma 2 to the first term in Eq. (33) and Lemma 3 to the second term yields

$$\begin{aligned}\mathbb{E}[\Phi_{t+1} \mid \mathcal{F}_t] \le V(\boldsymbol{y}_t) - V^* &- \frac{\alpha_t}{2}\|\nabla V(\boldsymbol{\theta}_t)\|^2 \\ &+ \left(Lc_1\alpha_t^2 - \frac{\alpha_t}{2}\right)\|\nabla V(\boldsymbol{z}_t)\|^2 \\ &+ \frac{Lc_{\beta,t}^2}{2c_1}\|\boldsymbol{v}_t\|^2 + \frac{\alpha_t L^2\rho_t^2}{2} + \frac{L\alpha_t^2 c_2}{2} \\ &+ \frac{Lc_{\beta,t}^2}{2c_1(1-\beta)}\left(\beta\|\boldsymbol{v}_t\|^2 + \frac{c_1\|\nabla V(\boldsymbol{z}_t)\|^2 + c_2}{1-\beta}\right).\end{aligned} \tag{34}$$

We first collect the coefficients of $\|\boldsymbol{v}_t\|^2$. By some calculations, we get

$$\begin{aligned}\frac{Lc_{\beta,t}^2}{2c_1} + \frac{L\beta c_{\beta,t}^2}{2c_1(1-\beta)} &= \frac{Lc_{\beta,t}^2}{2c_1}\left(1 + \frac{\beta}{1-\beta}\right) \\ &= \frac{Lc_{\beta,t}^2}{2c_1(1-\beta)}.\end{aligned} \tag{35}$$

Thus these terms combine with $V(\boldsymbol{y}_t) - V^*$ to give Φ_t. Next, we collect the coefficients of $\|\nabla V(\boldsymbol{z}_t)\|^2$ in Eq. (34), which gives

$$Lc_1\alpha_t^2 - \frac{\alpha_t}{2} + \frac{Lc_{\beta,t}^2}{2(1-\beta)^2}. \tag{36}$$

The remaining terms involving c_2 are $\frac{L\alpha_t^2 c_2}{2} + \frac{Lc_{\beta,t}^2 c_2}{2c_1(1-\beta)^2}$. Consequently, we have

$$\begin{aligned}\mathbb{E}[\Phi_{t+1} \mid \mathcal{F}_t] \le \Phi_t &- \frac{\alpha_t}{2}\|\nabla V(\boldsymbol{\theta}_t)\|^2 \\ &+ \left(Lc_1\alpha_t^2 - \frac{\alpha_t}{2} + \frac{Lc_{\beta,t}^2}{2(1-\beta)^2}\right)\|\nabla V(\boldsymbol{z}_t)\|^2 \\ &+ \frac{\alpha_t L^2\rho_t^2}{2} + \frac{L\alpha_t^2 c_2}{2} + \frac{Lc_{\beta,t}^2 c_2}{2c_1(1-\beta)^2}.\end{aligned} \tag{37}$$

By Lemma 5 and the step-size condition,

$$\alpha_t \le \frac{\eta_t}{1-\beta} \le \frac{(1-\beta)^2}{L\big(\beta^2 + 2c_1(1-\beta)^2\big)}. \tag{38}$$

Since $c_{\beta,t} = \beta\alpha_t$, the coefficient in Eq. (36) satisfies

$$\begin{aligned} & Lc_1\alpha_t^2 - \frac{\alpha_t}{2} + \frac{Lc_{\beta,t}^2}{2(1-\beta)^2} \\ &= Lc_1\alpha_t^2 - \frac{\alpha_t}{2} + \frac{L\beta^2\alpha_t^2}{2(1-\beta)^2} \\ &= \frac{\alpha_t}{2}\left[L\alpha_t\left(2c_1 + \frac{\beta^2}{(1-\beta)^2}\right) - 1\right] \\ &\leq 0. \end{aligned} \tag{39}$$

Because $\|\nabla V(\boldsymbol{z}_t)\|^2 \geq 0$, the corresponding term in Eq. (37) can be discarded. Moreover, substituting $c_{\beta,t} = \beta\alpha_t$ into the above inequality, we get

$$\frac{L\alpha_t^2 c_2}{2} + \frac{Lc_{\beta,t}^2 c_2}{2c_1(1-\beta)^2} = \frac{Lc_2\alpha_t^2}{2}\left(1 + \frac{\beta^2}{c_1(1-\beta)^2}\right). \tag{40}$$

We therefore obtain

$$\begin{aligned} \mathbb{E}[\Phi_{t+1} \mid \mathcal{F}_t] \leq \Phi_t - \frac{\alpha_t}{2}\|\nabla V(\boldsymbol{\theta}_t)\|^2 + \frac{\alpha_t L^2\rho_t^2}{2} \\ + \frac{Lc_2\alpha_t^2}{2}\left(1 + \frac{\beta^2}{c_1(1-\beta)^2}\right). \end{aligned} \tag{41}$$

Taking total expectations and using the tower property,

$$\begin{aligned} \mathbb{E}[\Phi_{t+1}] \leq \mathbb{E}[\Phi_t] - \frac{\alpha_t}{2}\mathbb{E}\|\nabla V(\boldsymbol{\theta}_t)\|^2 + \frac{\alpha_t L^2\rho_t^2}{2} \\ + \frac{Lc_2\alpha_t^2}{2}\left(1 + \frac{\beta^2}{c_1(1-\beta)^2}\right). \end{aligned} \tag{42}$$

Here α_t and ρ_t are deterministic, so they can be taken outside the expectation. Summing Eq. (42) over $t = 0, \ldots, T-1$ yields

$$\begin{aligned} \mathbb{E}[\Phi_T] - \Phi_0 \leq & -\frac{1}{2}\sum_{t=0}^{T-1}\alpha_t\mathbb{E}\|\nabla V(\boldsymbol{\theta}_t)\|^2 \\ & + \frac{L^2}{2}\sum_{t=0}^{T-1}\alpha_t\rho_t^2 + \frac{Lc_2}{2}\left(1 + \frac{\beta^2}{c_1(1-\beta)^2}\right)\sum_{t=0}^{T-1}\alpha_t^2. \end{aligned} \tag{43}$$

The Lyapunov terms telescope because

$$\sum_{t=0}^{T-1}\left(\mathbb{E}[\Phi_{t+1}] - \mathbb{E}[\Phi_t]\right) = \mathbb{E}[\Phi_T] - \Phi_0. \tag{44}$$

Since $v_0 = 0$, we have $y_0 = \boldsymbol{\theta}_0, \Phi_0 = V(\boldsymbol{\theta}_0) - V^* = \Delta(\boldsymbol{\theta}_0)$ and $\Phi_T = V(\boldsymbol{y}_t) - V^* + \frac{Lc_{\beta,T}^2}{2c_1(1-\beta)}\|\boldsymbol{v}_t\|^2 \geq 0$. Rearranging Eq. (43), multiplying by two, and applying the conditions of theorem gives

$$\begin{aligned} \sum_{t=0}^{T-1}\alpha_t\mathbb{E}\|\nabla V(\boldsymbol{\theta}_t)\|^2 \leq 2\Delta(\boldsymbol{\theta}_0) + L^2\sum_{t=0}^{T-1}\alpha_t\rho_t^2 \\ + Lc_2\left(1 + \frac{\beta^2}{c_1(1-\beta)^2}\right)\sum_{t=0}^{T-1}\alpha_t^2. \end{aligned} \tag{45}$$

For every $t < T$, we have $\min_{0\leq s<T}\mathbb{E}\|\nabla V(\boldsymbol{\theta}_s)\|^2 \leq \mathbb{E}\|\nabla V(\boldsymbol{\theta}_t)\|^2$.

Multiplying by $\alpha_t > 0$ and summing gives

$$\left(\sum_{t=0}^{T-1} \alpha_t\right) \min_{0 \le s < T} \mathbb{E}\|\nabla V(\boldsymbol{\theta}_s)\|^2 \le \sum_{t=0}^{T-1} \alpha_t \mathbb{E}\|\nabla V(\boldsymbol{\theta}_t)\|^2. \tag{46}$$

Combining this inequality with Eq. (45) and dividing by $\sum_{t=0}^{T-1} \alpha_t > 0$ proves Eq. (19).

It remains to express the bound in terms of η_t. By Lemma 5, we can see $\eta_t \le \alpha_t \le \frac{\eta_t}{1-\beta}$. Combing these facts, we have

$$\sum_{t=0}^{T-1} \alpha_t \ge \sum_{t=0}^{T-1} \eta_t, \tag{47}$$

$$\sum_{t=0}^{T-1} \alpha_t^2 \le \frac{1}{(1-\beta)^2} \sum_{t=0}^{T-1} \eta_t^2, \tag{48}$$

Since $\rho_t^2 \ge 0$, we get $\sum_{t=0}^{T-1} \alpha_t \rho_t^2 \le \frac{1}{1-\beta} \sum_{t=0}^{T-1} \eta_t \rho_t^2$. Note that $\Delta(\boldsymbol{\theta}_0) \ge 0$, these inequalities imply

$$\frac{2\Delta(\boldsymbol{\theta}_0)}{\sum_{t=0}^{T-1} \alpha_t} \le \frac{2\Delta(\boldsymbol{\theta}_0)}{\sum_{t=0}^{T-1} \eta_t}, \tag{49}$$

$$\frac{\sum_{t=0}^{T-1} \alpha_t^2}{\sum_{t=0}^{T-1} \alpha_t} \le \frac{1}{(1-\beta)^2} \frac{\sum_{t=0}^{T-1} \eta_t^2}{\sum_{t=0}^{T-1} \eta_t}, \tag{50}$$

and

$$\frac{\sum_{t=0}^{T-1} \alpha_t \rho_t^2}{\sum_{t=0}^{T-1} \alpha_t} \le \frac{1}{1-\beta} \frac{\sum_{t=0}^{T-1} \eta_t \rho_t^2}{\sum_{t=0}^{T-1} \eta_t}. \tag{51}$$

Substituting these three inequalities into Eq. (19) proves Eq. (20). Finally, substituting $c_1 = 1$ and $c_2 = \sigma^2/B$ into Eq. (20) proves Eq. (21). □

E.3 Proof of Theorem 4

Proof of Theorem 4. By Lemma 8, we obtain $\nabla V(\boldsymbol{\theta}_t) \to \mathbf{0}$. The second part of Lemma 9, using the energy-entry condition, supplies a constant $\underline{\alpha} > 0$ such that $\min_s p_t^*(s) \ge \underline{\alpha}$ for all t. We may apply Eq. (92) to obtain

$$0 \le V^{\pi_{\boldsymbol{\theta}_t}}(\mu) - V^*(\mu) \le \frac{\sqrt{|\mathcal{S}||\mathcal{A}|}}{\underline{\alpha}(1-\gamma)\mu_{\min}} \|\nabla V(\boldsymbol{\theta}_t)\|.$$

The right-hand side converges to zero, which proves Eq. (22). Taking μ to be a point mass at each state yields componentwise value convergence.

Returning to Eq. (89), the convergence $\delta_t \to 0$ and the full support of μ imply $p_t^*(s) \to 1$ for each state. If the optimal action is unique, the optimal set is the singleton $\{a^*(s)\}$, which proves the stated policy convergence. The parameter sequence itself need not converge to a finite vector, because a deterministic softmax limit generally requires diverging logit differences. □

E.4 Helpful Lemmas

Lemma 1 (Smoothness)**.** *If we use the softmax parameterization with the discounted factor γ, then $V^{\pi_{\boldsymbol{\theta}}}(\mu)$ is $8/(1-\gamma)^3$-smooth, i.e.,*

$$V^{\pi_{\boldsymbol{\theta}}}(\mu) - V^{\pi_{\boldsymbol{\theta}'}}(\mu) \le \frac{8}{(1-\gamma)^3} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|. \tag{52}$$

Note that this is given as Lemma 55 in (Agarwal et al., 2021) or Lemma 7 in (Mei et al., 2020). The proof can be found in their works.

Lemma 2 (One-step descent for the auxiliary sequence)**.** *Under the standing conditions and Assumptions 1 and 2, define α_t, $c_{\beta,t}$, and $\boldsymbol{y}_t$ as in Lemma 5. Then we have*

$$\begin{aligned}\mathbb{E}[V(\boldsymbol{y}_{t+1}) \mid \mathcal{F}_t] \le V(\boldsymbol{y}_t) - \frac{\alpha_t}{2}\|\nabla V(\boldsymbol{\theta}_t)\|^2 \\ + \left(Lc_1\alpha_t^2 - \frac{\alpha_t}{2}\right)\|\nabla V(\boldsymbol{z}_t)\|^2 \\ + \frac{Lc_{\beta,t}^2}{2c_1}\|\boldsymbol{v}_t\|^2 + \frac{\alpha_t L^2\rho_t^2}{2} + \frac{L\alpha_t^2 c_2}{2}.\end{aligned} \tag{53}$$

Proof of Lemma 2. By Lemma 5, we have $\boldsymbol{y}_{t+1} - \boldsymbol{y}_t = -\alpha_t g_t$. Applying the descent Lemma 4, to $V(\cdot)$ at $\boldsymbol{y}_t$ and $\boldsymbol{y}_{t+1}$ gives the pathwise inequality

$$\begin{aligned}V(\boldsymbol{y}_{t+1}) &\le V(\boldsymbol{y}_t) + \langle \nabla V(\boldsymbol{y}_t), \boldsymbol{y}_{t+1} - \boldsymbol{y}_t\rangle + \frac{L}{2}\|\boldsymbol{y}_{t+1} - \boldsymbol{y}_t\|^2 \\ &= V(\boldsymbol{y}_t) - \alpha_t\langle \nabla V(\boldsymbol{y}_t), g_t\rangle + \frac{L\alpha_t^2}{2}\|g_t\|^2.\end{aligned} \tag{54}$$

The quantities $\boldsymbol{y}_t$, $\boldsymbol{z}_t$, and $\boldsymbol{\theta}_t$ are $\mathcal{F}_t$-measurable, and α_t is deterministic. Taking conditional expectations in Eq. (54) therefore yields

$$\begin{aligned}\mathbb{E}[V(\boldsymbol{y}_{t+1}) \mid \mathcal{F}_t] \le V(\boldsymbol{y}_t) - \alpha_t \langle \nabla V(\boldsymbol{y}_t), \mathbb{E}[g_t \mid \mathcal{F}_t]\rangle \\ + \frac{L\alpha_t^2}{2}\mathbb{E}[\|g_t\|^2 \mid \mathcal{F}_t].\end{aligned} \tag{55}$$

By calculations, we have,

$$\begin{aligned}\mathbb{E}[V(\boldsymbol{y}_{t+1}) \mid \mathcal{F}_t] \le V(\boldsymbol{y}_t) - \alpha_t\langle \nabla V(\boldsymbol{y}_t), \nabla V(\boldsymbol{z}_t)\rangle \\ + \frac{L\alpha_t^2 c_1}{2}\|\nabla V(\boldsymbol{z}_t)\|^2 + \frac{L\alpha_t^2 c_2}{2}.\end{aligned} \tag{56}$$

We next bound the inner-product term. Adding and subtracting $\nabla V(\boldsymbol{\theta}_t)$ gives

$$\begin{aligned}&-\alpha_t\langle \nabla V(\boldsymbol{y}_t), \nabla V(\boldsymbol{z}_t)\rangle \\ &= -\alpha_t\langle \nabla V(\boldsymbol{\theta}_t), \nabla V(\boldsymbol{z}_t)\rangle \\ &\quad + \alpha_t\langle \nabla V(\boldsymbol{\theta}_t) - \nabla V(\boldsymbol{y}_t), \nabla V(\boldsymbol{z}_t)\rangle.\end{aligned} \tag{57}$$

For any vectors a, b, since $-\langle a, b\rangle = -\frac{1}{2}\|a\|^2 - \frac{1}{2}\|b\|^2 + \frac{1}{2}\|a - b\|^2$, and Applying this identity to the first term in Eq. (57), we obtain

$$\begin{aligned}&-\alpha_t\langle \nabla V(\boldsymbol{\theta}_t), \nabla V(\boldsymbol{z}_t)\rangle \\ &= -\frac{\alpha_t}{2}\|\nabla V(\boldsymbol{\theta}_t)\|^2 - \frac{\alpha_t}{2}\|\nabla V(\boldsymbol{z}_t)\|^2 \\ &\quad + \frac{\alpha_t}{2}\|\nabla V(\boldsymbol{\theta}_t) - \nabla V(\boldsymbol{z}_t)\|^2.\end{aligned} \tag{58}$$

For any $c_0 > 0$, by Young's inequality we get

$$\langle a, b\rangle \le \frac{c_0}{2}\|a\|^2 + \frac{1}{2c_0}\|b\|^2. \tag{59}$$

Applying this inequality to the second term in Eq. (57), we obtain

$$\begin{aligned}&\alpha_t\langle \nabla V(\boldsymbol{\theta}_t) - \nabla V(\boldsymbol{y}_t), \nabla V(\boldsymbol{z}_t)\rangle \\ &\le \frac{\alpha_t c_0}{2}\|\nabla V(\boldsymbol{\theta}_t) - \nabla V(\boldsymbol{y}_t)\|^2 + \frac{\alpha_t}{2c_0}\|\nabla V(\boldsymbol{z}_t)\|^2.\end{aligned} \tag{60}$$

By L-smoothness and the conditions of the lemma, we have

$$\begin{aligned}\|\nabla V(\boldsymbol{\theta}_t) - \nabla V(\boldsymbol{z}_t)\|^2 &\le L^2\|\boldsymbol{\theta}_t - \boldsymbol{z}_t\|^2 \\ &\le L^2\rho_t^2.\end{aligned} \tag{61}$$

Similarly, since $\boldsymbol{y}_t = \boldsymbol{\theta}_t - c_{\beta,t}\boldsymbol{v}_t$, this leads to

$$\begin{aligned}\|\nabla V(\boldsymbol{\theta}_t) - \nabla V(\boldsymbol{y}_t)\|^2 &\le L^2\|\boldsymbol{\theta}_t - \boldsymbol{y}_t\|^2 \\ &= L^2 c_{\beta,t}^2\|\boldsymbol{v}_t\|^2.\end{aligned} \tag{62}$$

Combining Eq. (57), Eq. (58), Eq. (60), Eq. (61), and Eq. (62), we obtain

$$\begin{aligned}&-\alpha_t\langle\nabla V(\boldsymbol{y}_t), \nabla V(\boldsymbol{z}_t)\rangle \\ &\le -\frac{\alpha_t}{2}\|\nabla V(\boldsymbol{\theta}_t)\|^2 + \frac{\alpha_t}{2}\left(\frac{1}{c_0} - 1\right)\|\nabla V(\boldsymbol{z}_t)\|^2 \\ &\quad + \frac{\alpha_t c_0 L^2 c_{\beta,t}^2}{2}\|\boldsymbol{v}_t\|^2 + \frac{\alpha_t L^2\rho_t^2}{2}.\end{aligned} \tag{63}$$

Substituting this bound into Eq. (56) gives

$$\begin{aligned}\mathbb{E}[V(\boldsymbol{y}_{t+1}) \mid \mathcal{F}_t] \le V(\boldsymbol{y}_t) &- \frac{\alpha_t}{2}\|\nabla V(\boldsymbol{\theta}_t)\|^2 \\ &+ \left[\frac{\alpha_t}{2}\left(\frac{1}{c_0} - 1\right) + \frac{L\alpha_t^2 c_1}{2}\right]\|\nabla V(\boldsymbol{z}_t)\|^2 \\ &+ \frac{\alpha_t c_0 L^2 c_{\beta,t}^2}{2}\|\boldsymbol{v}_t\|^2 + \frac{\alpha_t L^2\rho_t^2}{2} + \frac{L\alpha_t^2 c_2}{2}.\end{aligned} \tag{64}$$

Let $c_0 = \frac{1}{\alpha_t L c_1} > 0$, then we have

$$\begin{aligned}\frac{\alpha_t}{2}\left(\frac{1}{c_0} - 1\right) + \frac{L\alpha_t^2 c_1}{2} &= \frac{\alpha_t}{2}(\alpha_t L c_1 - 1) + \frac{L\alpha_t^2 c_1}{2} \\ &= L c_1\alpha_t^2 - \frac{\alpha_t}{2},\end{aligned} \tag{65}$$

In addition, this implies $\frac{\alpha_t c_0 L^2 c_{\beta,t}^2}{2} = \frac{L c_{\beta,t}^2}{2c_1}$. Substituting these identities into Eq. (64) proves Eq. (53). □

Lemma 3 (Momentum second-moment bound)**.** *Under Assumptions 1 and 2, we have*

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{v}_{t+1}\|^2 \mid \mathcal{F}_t] &\le \beta\|\boldsymbol{v}_t\|^2 + \frac{1}{1-\beta}\mathbb{E}[\|g_t\|^2 \mid \mathcal{F}_t] \\ &\le \beta\|\boldsymbol{v}_t\|^2 + \frac{c_1\|\nabla V(\boldsymbol{z}_t)\|^2 + c_2}{1-\beta}.\end{aligned} \tag{66}$$

Proof of Lemma 3. By $\boldsymbol{v}_{t+1} = \beta\boldsymbol{v}_t + \boldsymbol{g}_t$ we have $\|\boldsymbol{v}_{t+1}\|^2 = \beta^2\|\boldsymbol{v}_t\|^2 + 2\beta\langle\boldsymbol{v}_t, \boldsymbol{g}_t\rangle + \|\boldsymbol{g}_t\|^2$. Since $1 - \beta > 0$, Young's inequality gives

$$2\langle\boldsymbol{v}_t, \boldsymbol{g}_t\rangle \le (1-\beta)\|\boldsymbol{v}_t\|^2 + \frac{1}{1-\beta}\|\boldsymbol{g}_t\|^2. \tag{67}$$

Multiplying by $\beta \ge 0$ and substituting into the above inequality, we obtain

$$\begin{aligned}\|\boldsymbol{v}_{t+1}\|^2 &\le \left(\beta^2 + \beta(1-\beta)\right)\|\boldsymbol{v}_t\|^2 + \left(1 + \frac{\beta}{1-\beta}\right)\|\boldsymbol{g}_t\|^2 \\ &= \beta\|\boldsymbol{v}_t\|^2 + \frac{1}{1-\beta}\|\boldsymbol{g}_t\|^2.\end{aligned} \tag{68}$$

Taking conditional expectations and using the $\mathcal{F}_t$-measurability of $\boldsymbol{v}_t$ gives

$$\mathbb{E}[\|\boldsymbol{v}_{t+1}\|^2 \mid \mathcal{F}_t] \le \beta\|\boldsymbol{v}_t\|^2 + \frac{1}{1-\beta}\mathbb{E}[\|\boldsymbol{g}_t\|^2 \mid \mathcal{F}_t]. \tag{69}$$

Applying Assumption 2 at $\boldsymbol{z}_t$ proves Eq. (66). □

Lemma 4 (Descent lemma)**.** *If $F(\cdot) : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable and L-smooth, then, for any $x, y \in \mathbb{R}^d$,*

$$F(y) \le F(x) + \langle \nabla F(x), y - x \rangle + \frac{L}{2}\|y - x\|^2. \tag{70}$$

Proof. Let $d := y - x$ and define

$$\varphi(s) := F(x + sd), \qquad s \in [0, 1]. \tag{71}$$

By the chain rule,

$$\varphi'(s) = \langle \nabla F(x + sd), d \rangle. \tag{72}$$

The fundamental theorem of calculus therefore yields

$$\begin{aligned} F(y) - F(x) &= \varphi(1) - \varphi(0) \\ &= \int_0^1 \langle \nabla F(x + sd), d \rangle \, ds \\ &= \langle \nabla F(x), d \rangle + \int_0^1 \langle \nabla F(x + sd) - \nabla F(x), d \rangle \, ds. \end{aligned} \tag{73}$$

By Cauchy–Schwarz and L-smoothness,

$$\begin{aligned} \langle \nabla F(x + sd) - \nabla F(x), d \rangle &\le \|\nabla F(x + sd) - \nabla F(x)\| \, \|d\| \\ &\le L\|sd\| \, \|d\| \\ &= Ls\|d\|^2. \end{aligned} \tag{74}$$

Substituting this inequality into Eq. (73) gives

$$\begin{aligned} F(y) - F(x) &\le \langle \nabla F(x), d \rangle + L\|d\|^2 \int_0^1 s \, ds \\ &= \langle \nabla F(x), d \rangle + \frac{L}{2}\|d\|^2. \end{aligned} \tag{75}$$

Using $d = y - x$ proves Eq. (70). □

Lemma 5 (Auxiliary sequence for variable step sizes)**.** *Let $\beta \in [0, 1)$, and let $\{\eta_t\}_{t \ge 0}$ be a deterministic, positive, nonincreasing sequence. Define $\alpha_t := \sum_{k=0}^{\infty} \beta^k \eta_{t+k}, c_{\beta,t} := \beta\alpha_t$ and $\boldsymbol{y}_t := \boldsymbol{\theta}_t - c_{\beta,t}\boldsymbol{v}_t$. Then α_t is finite and satisfies $\alpha_t = \eta_t + \beta\alpha_{t+1}, \eta_t \le \alpha_t \le \frac{\eta_t}{1-\beta}$, and $\alpha_{t+1} \le \alpha_t$. Under the update rule, the auxiliary sequence satisfies the exact identity*

$$\boldsymbol{y}_{t+1} = \boldsymbol{y}_t - \alpha_t \boldsymbol{g}_t. \tag{76}$$

Proof. Because the step sizes are nonincreasing, $\eta_{t+k} \le \eta_t$ for every $k \ge 0$. we have, $0 \le \beta^k \eta_{t+k} \le \beta^k \eta_t$. Convergence of the geometric series implies $\sum_{k=0}^{\infty} \beta^k \eta_t = \frac{\eta_t}{1-\beta} < \infty$.

Thus the series defining α_t converges. Its first term is η_t, and all remaining terms are nonnegative, so

$$\eta_t \le \sum_{k=0}^{\infty} \beta^k \eta_{t+k} \le \eta_t \sum_{k=0}^{\infty} \beta^k = \frac{\eta_t}{1 - \beta}. \tag{77}$$

This proves the first result. Separating the first term and reindexing the remaining convergent series, we obtain

$$\alpha_t = \eta_t + \sum_{k=1}^{\infty} \beta^k \eta_{t+k} = \eta_t + \beta \sum_{j=0}^{\infty} \beta^j \eta_{t+1+j} = \eta_t + \beta\alpha_{t+1}. \tag{78}$$

This proves $\alpha_t = \eta_t + \beta\alpha_{t+1}$. For every $k \ge 0$, nonincreasing step sizes imply $\eta_{t+1+k} \le \eta_{t+k}$. Multiplying by $\beta^k \ge 0$ and summing gives

$$\alpha_{t+1} = \sum_{k=0}^{\infty} \beta^k \eta_{t+1+k} \le \sum_{k=0}^{\infty} \beta^k \eta_{t+k} = \alpha_t. \tag{79}$$

This proves the second result. We now derive the auxiliary update. By the definitions of $c_{\beta,t}$ and α_t, we get

$$\eta_t + c_{\beta,t+1} = \eta_t + \beta\alpha_{t+1} = \alpha_t, \qquad \beta\alpha_t = c_{\beta,t}. \tag{80}$$

Using these identities and the original recursion, we get

$$\begin{aligned}
\boldsymbol{y}_{t+1} &= \theta_{t+1} - c_{\beta,t+1}\boldsymbol{v}_{t+1} = \boldsymbol{\theta}_t - \eta_t\boldsymbol{v}_{t+1} - c_{\beta,t+1}\boldsymbol{v}_{t+1} \\
&= \boldsymbol{\theta}_t - (\eta_t + c_{\beta,t+1})\boldsymbol{v}_{t+1} = \boldsymbol{\theta}_t - \alpha_t\boldsymbol{v}_{t+1} \\
&= \boldsymbol{\theta}_t - \alpha_t(\beta\boldsymbol{v}_t + \boldsymbol{g}_t) = \boldsymbol{\theta}_t - \beta\alpha_t\boldsymbol{v}_t - \alpha_t\boldsymbol{g}_t \\
&= \boldsymbol{\theta}_t - c_{\beta,t}\boldsymbol{v}_t - \alpha_t\boldsymbol{g}_t \\
&= \boldsymbol{y}_t - \alpha_t\boldsymbol{g}_t.
\end{aligned}$$

This proves Eq. (76). Finally, the entire step-size sequence is deterministic, so α_t and $c_{\beta,t}$ are deterministic. Since $\boldsymbol{\theta}_t$ and $\boldsymbol{v}_t$ are $\mathcal{F}_t$-measurable, $\boldsymbol{y}_t$ is also $\mathcal{F}_t$-measurable. Thus the auxiliary construction is compatible with the conditional expectations used in the preceding lemmas. □

Lemma 6 (Second moment at the perturbed point)**.** *Under Algorithm 1, if Assumption 1 holds, then we have*

$$\mathbb{E}\left\|\boldsymbol{g}_t^{SAM}\right\|^2 \le 4L^2\rho^2\alpha^2 2\big(2L^2\rho^2(1-\alpha)^2 + 1\big)\mathbb{E}\left\|\boldsymbol{g}_t\right\|^2. \tag{81}$$

Proof of Lemma 6. By adding and subtracting $\boldsymbol{g}_t$ and using $\|\boldsymbol{a}+\boldsymbol{b}\|^2 \le 2\|\boldsymbol{a}\|^2 + 2\|\boldsymbol{b}\|^2$, we obtain

$$\left\|\boldsymbol{g}_t^{\mathrm{SAM}}\right\|^2 \le 2\left\|\boldsymbol{g}_t^{\mathrm{SAM}} - \boldsymbol{g}_t\right\|^2 + 2\left\|\boldsymbol{g}_t\right\|^2. \tag{82}$$

By Assumption 1 and Eq. (24), we have

$$\left\|\boldsymbol{g}_t^{\mathrm{SAM}} - \boldsymbol{g}_t\right\|^2 \le L^2\rho^2\left\|(1-\alpha)\boldsymbol{g}_t + \alpha\boldsymbol{u}_t\right\|^2.$$

Another application of the same quadratic inequality, together with $\|\boldsymbol{u}_t\| \le 1$, yields

$$\left\|\boldsymbol{g}_t^{\mathrm{SAM}} - \boldsymbol{g}_t\right\|^2 \le 2L^2\rho^2(1-\alpha)^2\left\|\boldsymbol{g}_t\right\|^2 2L^2\rho^2\alpha^2.$$

Plugging this inequality into Eq. (82) and taking conditional expectation prove Eq. (81). □

Lemma 7 (Alignment with the true policy gradient)**.** *Under Assumption 1, for every $\beta > 0$,*

$$\mathbb{E}[\langle \boldsymbol{g}_t^{SAM}, \nabla V(\boldsymbol{\theta}_t)\rangle] \ge \left(1 - \frac{\beta}{2}\right)\|\nabla V(\boldsymbol{\theta}_t)\|^2 - \frac{L^2\rho^2}{\beta}\left(\alpha^2 + (1-\alpha)^2\mathbb{E}[\|\boldsymbol{g}_t\|^2]\right). \tag{83}$$

In particular, when $\rho > 0$, choosing $\beta = L\rho$ gives

$$\begin{aligned}
\mathbb{E}[\langle \boldsymbol{g}_t^{SAM}, \nabla V(\boldsymbol{\theta}_t)\rangle] \ge &\left(1 - \frac{L\rho}{2}\right)\|\nabla V(\boldsymbol{\theta}_t)\|^2 \\
&- L\rho\alpha^2 - L\rho(1-\alpha)^2\mathbb{E}_t\left\|\boldsymbol{g}_t\right\|^2.
\end{aligned} \tag{84}$$

Proof of Lemma 7. Since $\mathbb{E}[\langle \boldsymbol{g}_t, \nabla V(\boldsymbol{\theta}_t)\rangle] = \|\nabla V(\boldsymbol{\theta}_t)\|^2$, we have

$$\mathbb{E}[\langle \boldsymbol{g}_t^{\mathrm{SAM}}, \nabla V(\boldsymbol{\theta}_t)\rangle = \|\nabla V(\boldsymbol{\theta}_t)\|^2 + \mathbb{E}_t[\langle \boldsymbol{g}_t^{\mathrm{SAM}} - \boldsymbol{g}_t, \nabla V(\boldsymbol{\theta}_t)\rangle].$$

Using Young's inequality, we get

$$\langle \boldsymbol{g}_t^{\mathrm{SAM}} - \boldsymbol{g}_t, \nabla V(\boldsymbol{\theta}_t)\rangle \le \frac{1}{2\beta}\left\|\boldsymbol{g}_t^{\mathrm{SAM}} - \boldsymbol{g}_t\right\|^2 \frac{\beta}{2}\|\nabla V(\boldsymbol{\theta}_t)\|^2.$$

Recall that the proof of Lemma 6 has already established

$$\mathbb{E}_t\left\|\boldsymbol{g}_t^{\mathrm{SAM}} - \boldsymbol{g}_t\right\|^2 \le 2L^2\rho^2\left((1-\alpha)^2\mathbb{E}_t\left\|\boldsymbol{g}_t\right\|^2 + \alpha^2\right).$$

Combining these displays proves Eq. (83). Further, we can prove Eq. (84) by setting $\beta = L\rho$. □

Lemma 8 (Lyapunov descent and asymptotic stationarity). *Suppose that $V(\cdot)$ is bounded from below and has an L-Lipschitz gradient. Assume that the step sizes are non-increasing and satisfy*

$$0 < \underline{\eta} \le \eta_t \le \overline{\eta} < \frac{2(1-\beta)}{L}, \qquad \sum_{t=0}^{\infty} \eta_t \rho_t^2 < \infty. \tag{85}$$

Then we have

$$\sum_{t=0}^{\infty} \eta_t \left\|\boldsymbol{v}_{t+1}\right\|^2 < \infty, \qquad \boldsymbol{v}_t \longrightarrow \mathbf{0}, \qquad \nabla V(\boldsymbol{\theta}_t) \longrightarrow \mathbf{0}, \tag{86}$$

where $c_\eta := 1 - \beta - L\overline{\eta}/2 > 0$.

Proof of Lemma 8. The difference between the perturbed and unperturbed gradients will be denoted by $\boldsymbol{e}_t := \nabla V(\boldsymbol{z}_t) - \nabla V(\boldsymbol{\theta}_t)$. Lipschitz continuity of the gradient gives $\|\boldsymbol{e}_t\| \le L\rho_t$. The momentum recursion can therefore be written as $\nabla V(\boldsymbol{\theta}_t) = \boldsymbol{v}_{t+1} - \beta \boldsymbol{v}_t - \boldsymbol{e}_t$.

Applying the descent lemma to $\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t = -\eta_t \boldsymbol{v}_{t+1}$ gives

$$\begin{aligned} V(\boldsymbol{\theta}_{t+1}) \le V(\boldsymbol{\theta}_t) &- \eta_t \left\|\boldsymbol{v}_{t+1}\right\|^2 + \beta\eta_t \langle \boldsymbol{v}_t, \boldsymbol{v}_{t+1}\rangle \\ &+ \eta_t \langle \boldsymbol{e}_t, \boldsymbol{v}_{t+1}\rangle + \frac{L\eta_t^2}{2} \left\|\boldsymbol{v}_{t+1}\right\|^2. \end{aligned}$$

The momentum cross term is handled using $2\langle \boldsymbol{v}_t, \boldsymbol{v}_{t+1}\rangle = \|\boldsymbol{v}_t\|^2 + \|\boldsymbol{v}_{t+1}\|^2 - \|\boldsymbol{v}_{t+1} - \boldsymbol{v}_t\|^2$. Define $\Phi_t := V(\boldsymbol{\theta}_t) + \frac{\beta\eta_{t-1}}{2} \|\boldsymbol{v}_t\|^2$. Since $\eta_t \le \eta_{t-1}$, the kinetic term involving $\boldsymbol{v}_t$ is bounded by the one already contained in Φ_t. Furthermore, by Young's inequality, we get $\langle \boldsymbol{e}_t, \boldsymbol{v}_{t+1}\rangle \le c_\eta \|\boldsymbol{v}_{t+1}\|^2/2 + \|\boldsymbol{e}_t\|^2/(2c_\eta)$. Combining these observations gives the key descent estimate

$$\Phi_{t+1} \le \Phi_t - \frac{c_\eta \eta_t}{2} \left\|\boldsymbol{v}_{t+1}\right\|^2 - \frac{\beta\eta_t}{2} \left\|\boldsymbol{v}_{t+1} - \boldsymbol{v}_t\right\|^2 + \frac{L^2\eta_t}{2c_\eta} \rho_t^2. \tag{87}$$

To absorb the summable error, introduce

$$\Psi_t := \Phi_t + \frac{L^2}{2c_\eta} \sum_{k=t}^{\infty} \eta_k \rho_k^2. \tag{88}$$

Eq. (87) shows that $\{\Psi_t\}$ is non-increasing. It is bounded from below because V is bounded from below and the other terms are nonnegative. Summing the decrease of Ψ_t therefore proves the first statement in Eq. (86). Since $\eta_t \ge \underline{\eta} > 0$, we also obtain $\sum_t \|\boldsymbol{v}_{t+1}\|^2 < \infty$, and hence $\boldsymbol{v}_t \to \mathbf{0}$.

The last condition in Eq. (85), together with $\eta_t \ge \underline{\eta}$, implies $\rho_t \to 0$. Thus $\boldsymbol{e}_t \to \mathbf{0}$, and the identity $\nabla V(\boldsymbol{\theta}_t) = \boldsymbol{v}_{t+1} - \beta\boldsymbol{v}_t - \boldsymbol{e}_t$ proves gradient convergence. Finally, Ψ_t converges, its tail term vanishes, and the kinetic term in Φ_t vanishes because $\boldsymbol{v}_t \to \mathbf{0}$. It follows that $V(\boldsymbol{\theta}_t)$ also converges. □

The next lemma combines the Bellman gap argument with the non-uniform gradient-domination property used in softmax policy-gradient analyses. Before introducing the lemma, we give some notations and definitions, For every state, let $\mathcal{A}^*(s) := \{a \in \mathcal{A} : Q^*(s,a) = V^*(s)\}$, $p_t^*(s) := \sum_{a\in\mathcal{A}^*(s)} \pi_{\boldsymbol{\theta}_t}(a \mid s)$ and $\mathcal{S}_{\text{sub}} := \{s \in \mathcal{S} : \mathcal{A}^*(s) \ne \mathcal{A}\}$. For $s \in \mathcal{S}_{\text{sub}}$, let $\Delta_s := \min_{a\notin\mathcal{A}^*(s)} \left[Q^*(s,a) - V^*(s)\right] > 0, \Gamma_\mu := \min_{s\in\mathcal{S}_{\text{sub}}} \mu(s)\Delta_s$. Note that if $\mathcal{S}_{\text{sub}}$ is empty, every policy is optimal and no further argument is needed. We hence consider the nontrivial case, for which $\Gamma_\mu > 0$.

Lemma 9 (Optimal-action mass and non-uniform gradient domination). *Under Assumption 3, the following statements hold.*

1. *For every $t \ge 0$ and every $s \in \mathcal{S}_{\text{sub}}$,*

$$p_t^*(s) \ge 1 - \frac{\delta_t}{\mu(s)\Delta_s}. \tag{89}$$

If $s \notin \mathcal{S}_{\text{sub}}$, then $p_t^*(s) = 1$.

2. *Suppose there exists a finite $T \geq 1$ such that*

$$\mathcal{E}_T := \delta_T + \frac{\beta \eta_{T-1}}{2} \|\boldsymbol{v}_t\|^2 + \frac{L^2}{2c_\eta} \sum_{k=T}^{\infty} \eta_k \rho_k^2 < \Gamma_\mu. \tag{90}$$

Then the optimal-action mass is uniformly bounded away from zero:

$$\underline{\alpha} := \inf_{t \geq 0} \min_{s \in \mathcal{S}} p_t^*(s) > 0. \tag{91}$$

More precisely, one may take

$$\underline{\alpha} = \min \left\{ \min_{\substack{0 \leq t < T \\ s \in \mathcal{S}}} p_t^*(s),\ 1 - \frac{\mathcal{E}_T}{\Gamma_\mu} \right\}.$$

3. *If $\underline{\alpha} > 0$, then, for every initial-state distribution μ and every $t \geq 0$,*

$$0 \leq V^{\pi_{\boldsymbol{\theta}_t}}(\mu) - V^*(\mu) \leq \frac{\sqrt{|\mathcal{S}||\mathcal{A}|}}{\underline{\alpha}(1-\gamma)\mu_{\min}} \|\nabla F_\mu(\boldsymbol{\theta}_t)\|. \tag{92}$$

Proof of Lemma 9. We prove the three statements in order. For the first statement, let $\pi_t := \pi_{\boldsymbol{\theta}_t}$ and $\Delta V(\boldsymbol{\theta}_t) := V^{\pi_t} - V^*$ for convenience. Since the problem is a minimization problem, we have that $\Delta V(\boldsymbol{\theta}_t)$ is component-wise nonnegative. By the Bellman equation, we get

$$\Delta V(\boldsymbol{\theta}_t) = T^{\pi_t} V^* - V^* + \gamma P^{\pi_t} \Delta V(\boldsymbol{\theta}_t) \geq T^{\pi_t} V^* - V^*.$$

At state s, an optimal action contributes zero to the last Bellman gap, whereas every suboptimal action contributes at least Δ_s. It follows that $\Delta V(\boldsymbol{\theta}_t)(s) \geq \Delta_s(1 - p_t^*(s))$. On the other hand, $\delta_t = \sum_s \mu(s) \Delta V(\boldsymbol{\theta}_t)(s) \geq \mu(s) \Delta V(\boldsymbol{\theta}_t)(s)$. Combining these two inequalities proves Eq. (89).

We next prove the second statement. By Eq. (88), the corrected energy Ψ_t is non-increasing. Hence, for every $t \geq T$, we have

$$\begin{aligned} \delta_t &\leq \Phi_t - V^*(\mu) \leq \Psi_t - V^*(\mu) \leq \Psi_T - V^*(\mu) \\ &= \delta_T + \frac{\beta \eta_{T-1}}{2} \|\boldsymbol{v}_t\|^2 + \frac{L^2}{2c_\eta} \sum_{k=T}^{\infty} \eta_k \rho_k^2. \end{aligned}$$

Let the final quantity be B_T. Condition in Eq. (90) states exactly that $B_T < \Gamma_\mu$. Using the first part of this lemma gives $p_t^*(s) \geq 1 - B_T/\Gamma_\mu > 0$ for all $t \geq T$. For the finite timepoint $0 \leq t < T$, every finite softmax parameter assigns strictly positive probability to every action. Taking the smaller of the tail bound and the finite-prefix minimum produces the desired constant $\underline{\alpha}$.

For the final statement, define a policy supported on the Bellman-optimal action set by

$$\pi_t^*(a \mid s) := \begin{cases} \pi_{\boldsymbol{\theta}_t}(a \mid s)/p_t^*(s), & a \in \mathcal{A}^*(s), \\ 0, & a \notin \mathcal{A}^*(s). \end{cases}$$

Every action in the support of π_t^* satisfies the Bellman optimality equation, so π_t^* is an optimal policy. The softmax policy-gradient identity for the cost objective is

$$\frac{\partial V(\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}(s,a)} = \frac{1}{1-\gamma} d_\mu^{\pi_t}(s) \pi_t(a \mid s) A^{\pi_t}(s,a), \tag{93}$$

where $A^\pi(s,a) := Q^\pi(s,a) - V^\pi(s)$. Applying the relation between the ℓ_1 and ℓ_2 norms, retaining only optimal actions, and then factoring out $p_t^*(s)$ gives

$$\begin{aligned} \|\nabla V(\boldsymbol{\theta}_t)\| \geq{}& \frac{\min_s p_t^*(s)}{(1-\gamma)\sqrt{|\mathcal{S}||\mathcal{A}|}\, \kappa_t(\mu)} \\ &\times \sum_s d_\mu^{\pi_t^*}(s) \sum_a \pi_t^*(a \mid s) |A^{\pi_t}(s,a)|, \end{aligned}$$

where $\kappa_t(\mu) := \left\| d_\mu^{\pi_t^*} / d_\mu^{\pi_t} \right\|_\infty$. The performance-difference lemma shows that the final double sum divided by $1-\gamma$ is at least $V^{\pi_t}(\mu) - V^*(\mu)$.

It remains to control the distribution mismatch. since the time-zero term is present in the discounted occupancy measure, we have $d_\mu^{\pi_t}(s) \geq (1-\gamma)\mu(s)$. Since a normalized occupancy measure is at most one, this implies $\kappa_t(\mu) \leq [(1-\gamma)\mu_{\min}]^{-1}$. Substituting this estimate and $\min_s p_t^*(s) \geq \underline{\alpha}$ proves Eq. (92). □

F Implementation of the Experiments

In this section, we provide detailed information about the environments and perturbation protocols, model architectures, hyperparameters, and experimental settings used in our experiments. Our implementation builds on the experimental framework of Lee & Yoon (2025), with the settings and method-specific differences detailed below.

Environments and Perturbation Protocols. We consider three continuous-control tasks from Gymnasium (Towers et al., 2025) with the MuJoCo (Todorov et al., 2012) physics simulator:

- `HalfCheetah-v4`: The half-cheetah is a planar biped robot with 8 rigid links, including two legs and a torso, along with 6 actuated joints. The 17D state space includes joint angles and joint velocities. The adversary applies a 6D action with 2D forces on the torso and both feet in order to destabilize it.
- `Hopper-v4`: The hopper is a planar monopod robot with 4 rigid links, corresponding to the torso, upper leg, lower leg, and foot, along with 3 actuated joints. The 11D state space includes joint angles and joint velocities. The adversary applies a 2D force on the foot.
- `Walker2D-v4`: The walker is a planar biped robot consisting of 7 links, corresponding to two legs and a torso, along with 6 actuated joints. The 17D state space includes joint angles and joint velocities. The adversary applies a 4D action with 2D forces on both feet.

Following the evaluation framework of Lee & Yoon (2025), we examine robustness to action perturbations, changes in physical parameters, and noisy reward feedback:

- **Action perturbations:** During evaluation, we add independent Gaussian noise $\epsilon_t \sim \mathcal{N}(0, \sigma_a^2 I)$ to the deterministic mean action at each time step and clip the resulting action to the environment's action bounds. We evaluate 11 equally spaced noise levels with $\sigma_a \in [0, 0.5]$ to assess sensitivity to errors in action execution.
- **transition dynamics perturbations:** We modify torso mass and foot-contact friction during evaluation to assess robustness to changes in environment dynamics. Torso mass is scaled by 11 equally spaced factors in $[0.5, 1.5]$. For the selected foot geometries, the first two components of the MuJoCo friction vector are scaled by 11 equally spaced factors in $[0.4, 1.6]$, while the third component is fixed at 0.1. We evaluate mass and friction variations separately and jointly, with the joint evaluation covering all 11×11 combinations.
- **Reward perturbations:** In a separate training condition, we corrupt the reward at each time step by adding independent Gaussian noise $\xi_t \sim \mathcal{N}(0, 0.1^2)$ before reward scaling. The resulting policies are evaluated in the nominal environment without additional reward noise, measuring their ability to learn effective policies from noisy reward feedback.

Model Architectures. For all methods, the actor (policy network) and critic (value network) are separate multilayer perceptrons (MLPs), each with two hidden layers of 64 units and `Tanh` activations. The actor parameterizes a diagonal Gaussian policy with a state-dependent mean and a learned, state-independent log standard deviation, while the critic outputs a scalar state-value estimate. We apply orthogonal initialization to all linear layers. For RARL, the protagonist and adversary use separate actor–critic pairs with the same architecture.

Baselines. We consider the following baselines, covering robust RL methods and PPO variants with alternative optimization methods:

- **RNAC** (Zhou et al., 2023) combines robust policy evaluation with natural policy gradient updates to learn policies that are robust to uncertainty in transition dynamics.
- **RARL** (Pinto et al., 2017) jointly trains a control policy and an adversary that applies disturbance forces, encouraging robustness through a zero-sum minimax objective.
- **PPO+SAM** (Lee & Yoon, 2025) integrates sharpness-aware minimization into PPO to seek flatter reward landscapes in policy parameter space and improve robustness to environmental perturbations.
- **PPO+NAG** is a PPO variant implemented in this work that uses Nesterov's accelerated gradient (Chen et al., 2023) for optimization, serving as a momentum-based baseline.

Hyperparameters. Following the experimental setup of (Engstrom et al., 2019; Lee & Yoon, 2025), we adopt a common PPO backbone and shared training settings to facilitate fair comparisons across methods. The basic PPO settings for each environment are summarized in Table 4. We perform grid searches over the method-specific hyperparameters. For PPO+SAM and PPO+mSAM, we tune the perturbation radius ρ over the same candidate set $\{0.001, 0.005, 0.008, 0.01, 0.02, 0.05\}$. For RARL, we tune the adversarial action coefficient α_{adv} over $\{0.10, 0.15, 0.25, 0.30\}$ and alternate between one protagonist rollout phase and one adversary rollout phase. To evaluate Nesterov accelerated gradient while retaining the adaptive optimization framework used by the other PPO variants, we employ NAG (Dozat, 2016), which incorporates Nesterov momentum into Adam's adaptive updates. We refer to this baseline as PPO+NAG and use the same PPO backbone and actor and critic learning-rate settings to facilitate a controlled comparison. The gradient clipping setting in the table applies to all methods except PPO+SAM, whose implementation does not apply gradient clipping within its SAM updates.

Table 4: Hyperparameters for three continuous-control environments.

Hyperparameter	**HalfCheetah-v4**	**Hopper-v4**	**Walker2d-v4**
Batch size	2048	2048	2048
Minibatch	64	64	64
Policy distribution	Gaussian	Gaussian	Gaussian
Discount factor γ	0.99	0.99	0.99
GAE parameter λ	0.95	0.95	0.95
PPO epochs per update $K = 10$	10	10	10
PPO Clipping ε	0.2	0.2	0.2
Policy network LR	0.0003	0.0003	0.0003
Value network LR	0.0003	0.0003	0.0003
LR schedule	Linear decay	Linear decay	Linear decay
Optimizer	Adam	Adam	Adam
Entropy coefficient	0.01	0.01	0.01
Maximum gradient norm	0.5	0.5	0.5
Maximum training steps	2×10^6	3×10^6	2×10^6
Evaluation interval (steps)	5000	5000	5000

G Additional Experiments and Setup Details

G.1 Experiments on Discrete Action Environments

In addition to the continuous-control tasks reported in the main text, we evaluate our method on two discrete action environments provided by Gymnasium (Towers et al., 2025), `CartPole-v1` and `LunarLander-v3`, to assess its applicability to discrete action spaces. These tasks involve balancing a pole on a moving cart and controlling a lunar lander to land safely on a designated pad, respectively.

Experimental Setup Following the evaluation framework of Lee & Yoon (2025), we compare PPO, PPO+SAM, PPO+NAG, and PPO+mSAM under action and reward perturbations. The basic PPO hyperparameters follow those used in the continuous-control experiments. Both the actor and critic retain two hidden layers of 64 units with Tanh activations, while the actor uses a softmax output to parameterize a categorical action distribution. For PPO+SAM and PPO+mSAM, we consider the same perturbation radius $\rho \in \{0.001, 0.005, 0.008, 0.01\}$. Each configuration is evaluated using five training seeds.

Action Robustness Evaluation During evaluation, we add independent zero-mean Gaussian noise to the policy logits before the softmax operation and select the action with the highest resulting probability. We vary the noise standard deviation σ_a over 11 equally spaced values in $[0, 0.5]$ and report the mean episodic return over 100 evaluation episodes for each trained policy at each noise level. As shown in Fig. 5, PPO+mSAM achieves the highest mean return across all evaluated noise levels in both environments. These results further demonstrate the effectiveness of PPO+mSAM in maintaining policy performance under action-selection perturbations in discrete-action environments.

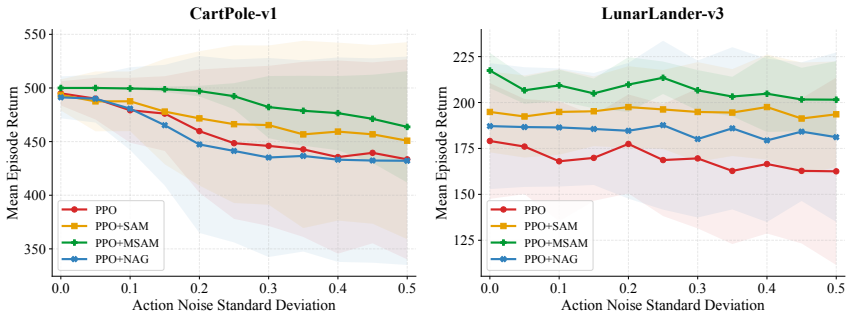


Figure 5: **Action robustness in discrete-action environments.** In `CartPole-v1`, the methods perform similarly without perturbations, while PPO+mSAM preserves near-nominal returns over a wider noise range and exhibits a smaller overall performance decline. In `LunarLander-v3`, PPO+mSAM consistently achieves the highest mean return despite fluctuations across noise levels. These results complement the continuous-control findings, supporting the robustness of PPO+mSAM to perturbations in discrete action selection.

Reward Robustness Evaluation To assess the effect of noisy reward feedback on policy learning, we add independent zero-mean Gaussian noise to the rewards during training, with standard deviations $\sigma_r \in \{0.0, 0.1, 0.2, 0.5, 1.0\}$. We then evaluate the trained policies in the nominal environment without reward noise and report the mean episodic return and standard deviation over five seeds. This protocol measures whether policies learned from noisy rewards retain their performance under nominal evaluation conditions. As shown in Table 5, PPO+mSAM maintains strong nominal performance when trained with low to moderate reward noise in both environments. These results support its effectiveness in learning from noisy reward feedback, although its advantage depends on the environment and noise level.

Table 5: **Reward robustness in discrete-action environments.** PPO+mSAM maintains consistently high returns across all reward noise levels in `CartPole-v1`, where all methods perform similarly. In `LunarLander-v3`, PPO+mSAM consistently outperforms PPO+SAM and maintains strong performance under low to moderate noise. The best and second-best results are bold and underlined, respectively.

	Method	$\sigma_r = 0.0$	$\sigma_r = 0.1$	$\sigma_r = 0.2$	$\sigma_r = 0.5$	$\sigma_r = 1.0$
CartPole	PPO	495.8 ± 9.5	498.6 ± 3.1	497.7 ± 4.3	498.6 ± 3.2	**500.0** ± 0.0
	PPO+SAM	<u>497.5</u> ± 5.3	498.0 ± 4.6	498.0 ± 4.5	<u>498.6</u> ± 3.1	**500.0** ± 0.0
	PPO+mSAM	**500.0** ± 0.0	**500.0** ± 0.0	**500.0** ± 0.0	**500.0** ± 0.0	**500.0** ± 0.0
	PPO+NAG	496.2 ± 8.5	<u>500.0</u> ± 0.0	<u>499.1</u> ± 1.6	497.4 ± 5.8	**500.0** ± 0.0
LunarLander	PPO	172.3 ± 22.4	**210.2** ± 20.8	191.3 ± 35.3	179.0 ± 50.1	**192.8** ± 38.0
	PPO+SAM	<u>197.4</u> ± 18.2	180.5 ± 28.3	167.9 ± 36.0	186.7 ± 32.0	163.6 ± 52.4
	PPO+mSAM	**212.2** ± 6.0	201.4 ± 58.0	<u>200.4</u> ± 34.1	<u>206.5</u> ± 32.6	169.1 ± 49.9
	PPO+NAG	189.8 ± 29.5	<u>207.1</u> ± 22.0	**201.6** ± 45.9	**207.3** ± 12.8	<u>182.1</u> ± 28.9

G.2 Extension to Other Policy Learning Algorithms

To examine the applicability of mSAM beyond PPO, we incorporate it into two off-policy algorithms: Soft Actor-Critic (SAC) and Twin Delayed Deep Deterministic Policy Gradient (TD3). The resulting variants, SAC+mSAM and TD3+mSAM, allow us to investigate whether the benefits of mSAM extend to off-policy learning with stochastic and deterministic policies, respectively.

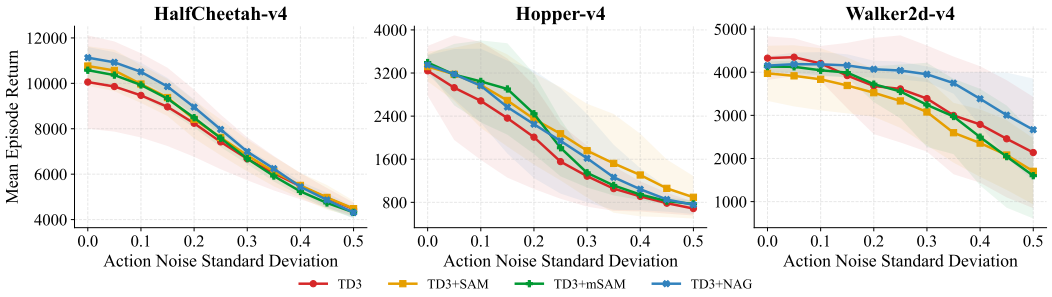


Figure 6: Action robustness evaluation of TD3 and its variants.

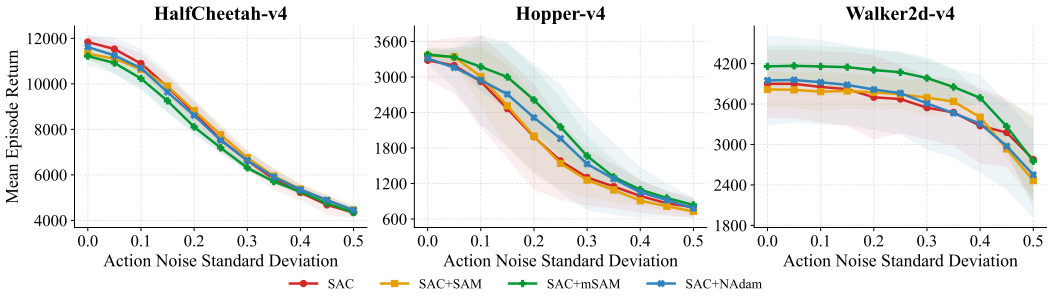


Figure 7: Action robustness evaluation of SAC and its variants.

Experimental Setup. We evaluate SAC, TD3, and their SAM, NAG, and mSAM variants on the three continuous-control environments used in the main text. Within each algorithm, all variants share the same network architectures and base training settings. Both actor and critic networks have two 256-unit hidden layers with ReLU activations. We use a learning rate of 3×10^{-4} for both networks, a minibatch size of 256, and a discount factor of 0.99. For SAM and mSAM, we consider perturbation radii $\rho \in 0.005, 0.008, 0.01, 0.05$. All experiments use five random seeds and follow the main-text protocols for action perturbations, mass and friction variations, and noisy reward feedback.

Robustness under Action Perturbations. Following the evaluation protocol described in our PPO experiments, we compare the TD3 variants under action noise in Fig. 6. TD3+mSAM yields modest gains over TD3 on HalfCheetah-v4 and Hopper-v4 at low-to-moderate noise levels, but achieves lower returns on Walker2d-v4 under stronger noise. The gains from mSAM are concentrated at low-to-moderate noise levels, with no consistent advantage over TD3 under stronger action perturbations. As shown in Fig. 7, SAC+mSAM shows clearer benefits on Hopper-v4 and Walker2d-v4, with limited gains on HalfCheetah-v4. Although improvements are less consistent than in the PPO experiments, these results suggest that mSAM can also benefit other policy optimization algorithms.

Robustness under Transition Perturbations. We evaluate the TD3 variants under friction and torso-mass variations using the same protocol as in our PPO experiments. As shown in Fig. 8, TD3+mSAM achieves higher returns than TD3 across the tested friction factors on HalfCheetah-v4 and at several settings on Hopper-v4. On Walker2d-v4, its advantage is limited to low friction, while higher friction leads to a larger performance drop than TD3. Fig. 9 shows that SAC+mSAM consistently achieves the highest mean returns on Walker2d-v4, while underperforming SAC on HalfCheetah-v4. On Hopper-v4, both variants provide gains near nominal friction but remain sensitive to larger deviations. These results show that mSAM offers benefits beyond PPO, although the improvements depend on the environment and perturbation level.

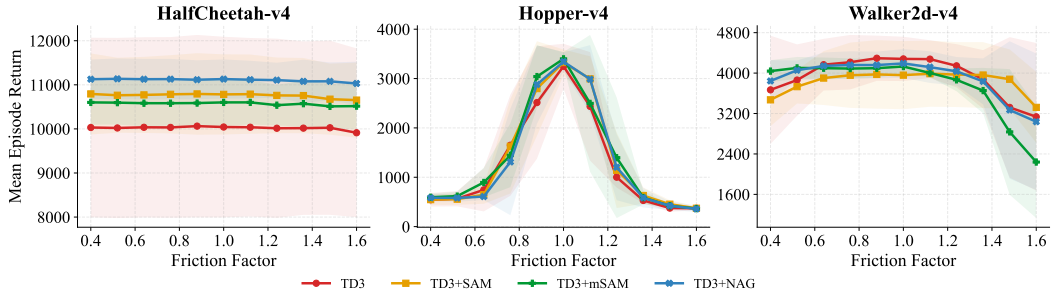


Figure 8: Robustness evaluation of TD3 and its variants under friction variations.

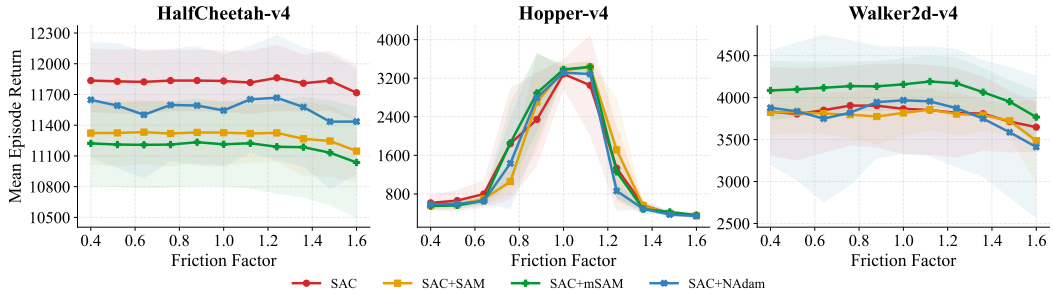


Figure 9: Robustness evaluation of SAC and its variants under friction variations.

Next, we evaluate robustness to torso-mass variations while keeping friction fixed. As shown in Fig. 10, TD3+mSAM achieves higher returns than TD3 across a range of mass factors, particularly on HalfCheetah-v4 and on Hopper-v4 under increased torso mass. However, these gains do not translate into a consistent advantage over TD3+SAM. Fig. 11 shows that SAC+mSAM achieves the highest mean returns throughout the evaluated range on Walker2d-v4 and shows benefits under increased mass on Hopper-v4, while generally underperforming SAC on HalfCheetah-v4. These results further support the applicability of mSAM beyond PPO, with environment-dependent benefits.

Finally, Figs. 12 and 13 compares TD3 and SAC variants under joint torso-mass and friction variations. TD3+mSAM generally improves over TD3 on HalfCheetah-v4 but yields lower returns at high friction on Walker2d-v4. SAC+mSAM maintains higher returns across much of the evaluated grid on Walker2d-v4, with limited gains on HalfCheetah-v4. On Hopper-v4, both mSAM variants improve

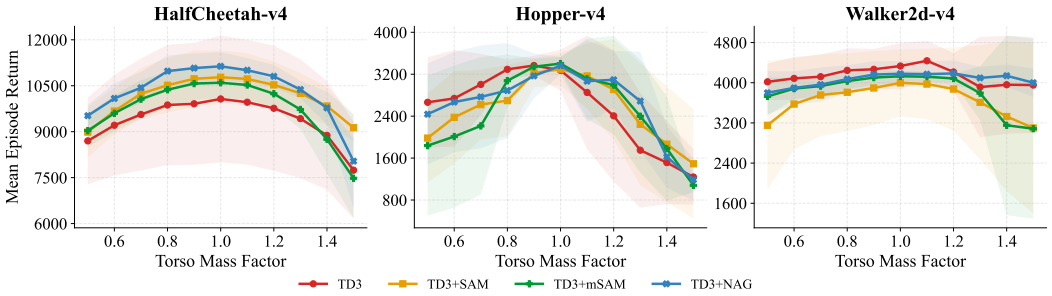


Figure 10: Robustness evaluation of TD3 and its variants under torso-mass variations.

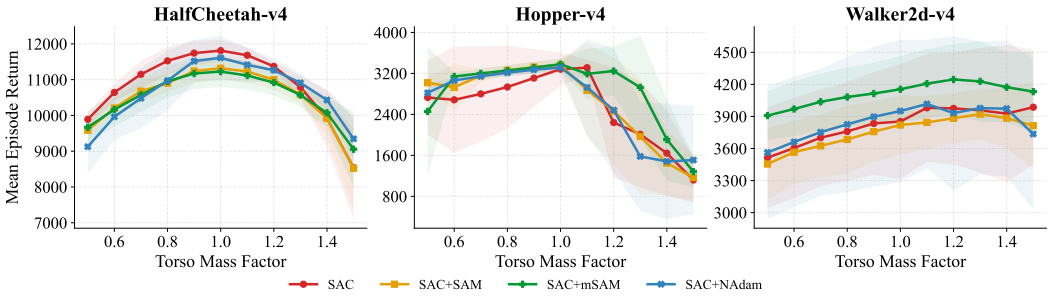


Figure 11: Robustness evaluation of SAC and its variants under torso-mass variations.

performance under some parameter combinations, although all methods remain sensitive to large friction changes.

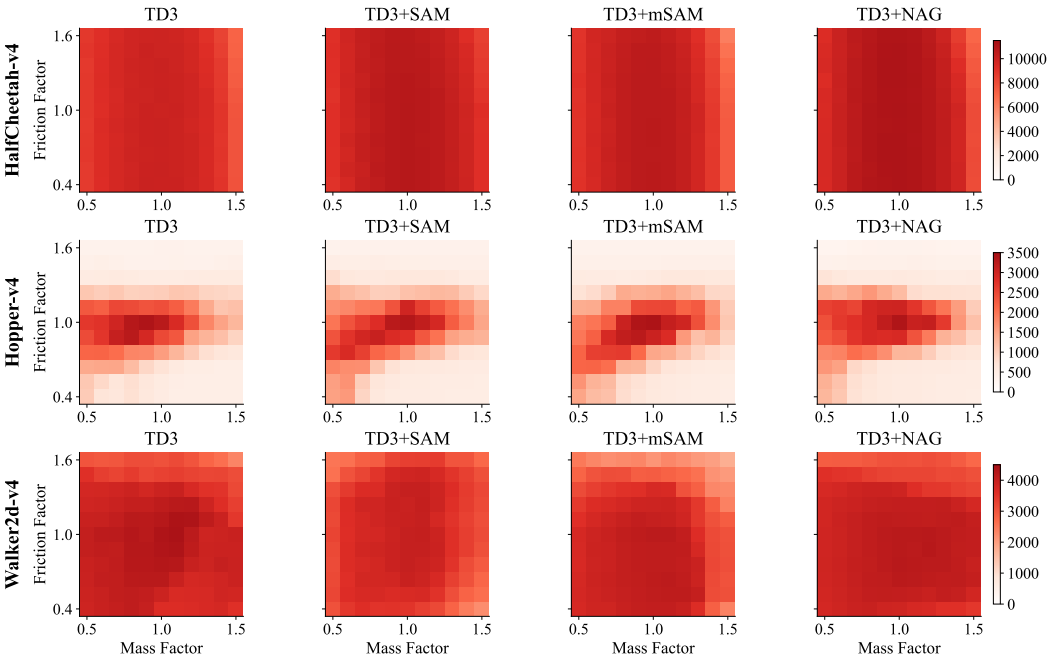


Figure 12: Robustness of TD3 and its variants under joint torso-mass and friction variations.

Robustness under Reward Perturbations. We train TD3 and its variants with either noise-free rewards or rewards perturbed by zero-mean Gaussian noise ($\sigma_r = 0.1$). All policies are evaluated in

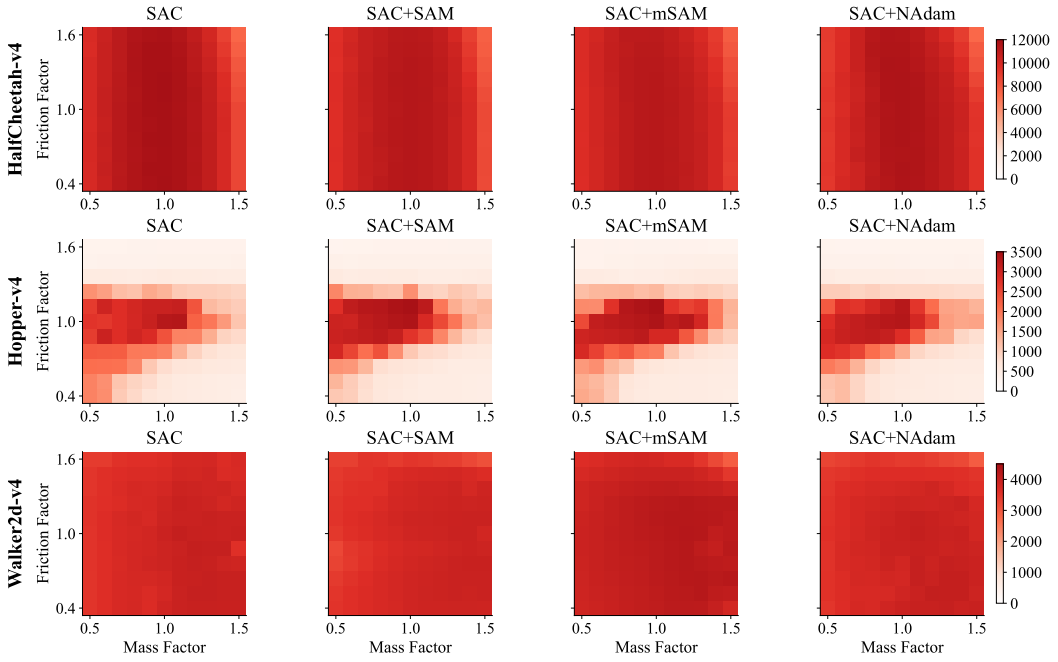


Figure 13: Robustness of SAC and its variants under joint torso-mass and friction variations.

the nominal environment without test-time noise, and results are averaged over five training seeds. As shown in Table 6, TD3+mSAM achieves the highest mean return on Hopper-v4 and Walker2d-v4 under noisy-reward training and ranks second on HalfCheetah-v4. In this setting, it yields higher mean returns than both TD3 and TD3+SAM across all three tasks.

Table 6: **Performance of TD3 and its variants trained without (Nominal) and with (Noisy, $\sigma_r = 0.1$) reward noise.** Results are mean returns over five training seeds, evaluated without test-time noise. SAM and mSAM use $\rho = 0.01$. Parentheses show changes from Nominal to Noisy; bold marks the highest mean in each column.

	HalfCheetah-v4		Hopper-v4		Walker2d-v4	
Algorithm	Nominal	Noisy	Nominal	Noisy	Nominal	Noisy
TD3	10056	9911 (−146)	3241	3276 (+35)	**4327**	4046 (−281)
TD3+NAG	**11132**	**10552 (−579)**	3351	3252 (−99)	4154	3999 (−155)
TD3+SAM	10763	10103 (−660)	3320	3085 (−234)	3975	3875 (−100)
TD3+mSAM	10579	10291 (−288)	**3395**	**3370 (−24)**	4131	**4119 (−12)**

G.3 Sensitivity Studies

The perturbation radius ρ controls the size of the parameter neighborhood considered by mSAM during policy optimization. Its choice can therefore affect both nominal policy performance and robustness to action perturbations. To examine this effect, we evaluate PPO+mSAM with $\rho \in \{0.001, 0.005, 0.008, 0.01, 0.02, 0.05\}$ in `HalfCheetah-v4`, `Hopper-v4`, and `Walker2d-v4`, varying the action noise standard deviation σ_a from 0 to 0.5 during evaluation. As shown in Fig. 14, the effect of ρ depends on both the environment and the noise level. These results suggest that, within the tested range, a larger perturbation radius can improve robustness to strong action noise, although it may reduce nominal performance in some environments.

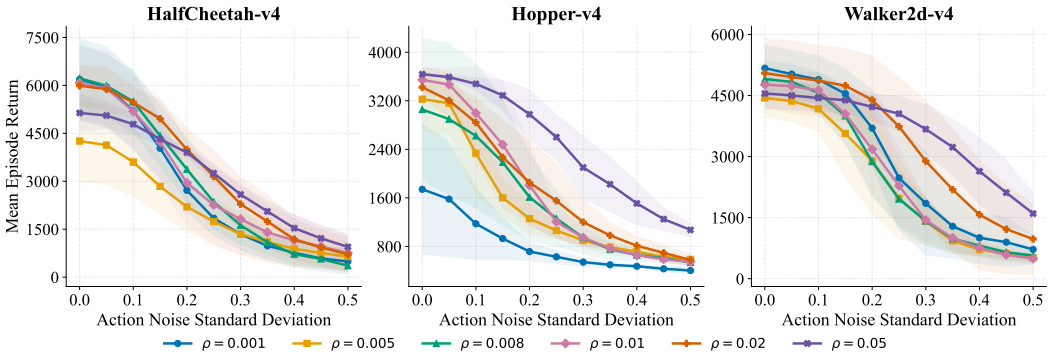


Figure 14: **Sensitivity of PPO+mSAM to the perturbation radius ρ.** In `HalfCheetah-v4` and `Walker2d-v4`, smaller radii generally yield higher returns under nominal or low-noise conditions, whereas $\rho = 0.05$ achieves the highest returns under stronger perturbations. In `Hopper-v4`, $\rho = 0.05$ performs best throughout the evaluated noise range.

G.4 Detailed Computational Efficiency Analysis

Training Time per Run. Integrating SAM into PPO increases computational cost because each policy update requires an additional gradient evaluation to construct the parameter perturbation. To assess the computational efficiency of PPO+mSAM, we compare the wall-clock time required for a single training run across three environments. Table 7 reports the training times of PPO+mSAM and the baseline methods. PPO+mSAM consistently requires less training time than PPO+SAM, demonstrating its computational advantage across the evaluated environments. Together with the robustness results, this comparison shows that PPO+mSAM offers a favorable balance between computational efficiency and policy robustness.

Table 7: **Wall-clock time in seconds for a single training run.** PPO+mSAM reduces training time by approximately 21-45% compared with PPO+SAM across the three environments. Its runtime is close to that of PPO in `HalfCheetah-v4` and `Hopper-v4`, with a larger overhead in `Walker2d-v4`.

Methods	HalfCheetah-v4	Hopper-v4	Walker2d-v4
PPO	726.36	1384.32	799.04
PPO+NAG	877.93	1222.90	934.03
PPO+SAM	1419.03	1778.14	1305.40
PPO+mSAM	778.16	1321.04	1026.58
RARL	806.00	1080.96	938.78
RNAC	807.38	1075.99	821.13

Time to Target Performance. Beyond the computational cost of a complete training run, we examine how quickly each method reaches a given level of policy performance. We plot evaluation returns against cumulative update time and measure the time at which each method first reaches a predefined return threshold. The thresholds are set to 5000, 3000, and 4000 for `HalfCheetah-v4`, `Hopper-v4`, and `Walker2d-v4`, respectively.

As shown in Fig. 15, PPO+mSAM reaches the target return first in both `HalfCheetah-v4` and `Hopper-v4`. In `Walker2d-v4`, PPO reaches the target earlier, while PPO+mSAM still requires less update time than PPO+SAM. Moreover, PPO+SAM does not reach the target in `HalfCheetah-v4` within the plotted time range. These results complement the training-time comparison, showing that PPO+mSAM can also reduce the computation required to attain a target return compared with PPO+SAM.

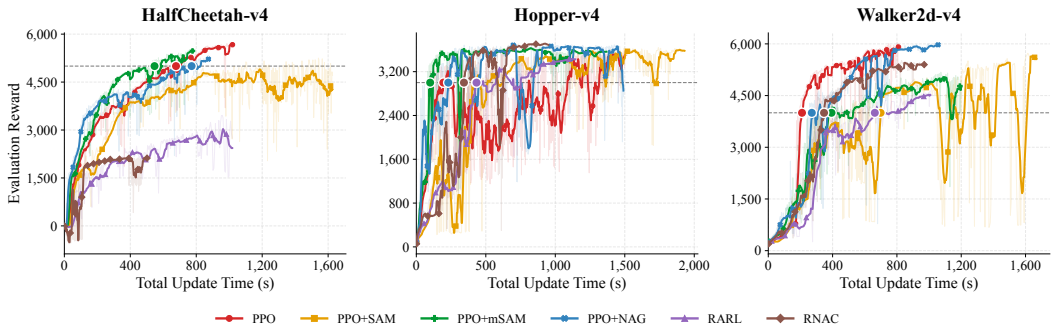


Figure 15: **Evaluation return versus cumulative update time.** PPO+mSAM reaches the target first in `HalfCheetah-v4` and `Hopper-v4`, and earlier than PPO+SAM in `Walker2d-v4`.

G.5 Rewards Surface Visualizations and Flatness Measurements

We visualize the reward landscape following (Sullivan et al., 2022). To quantify its flatness, we compute two metrics: the largest eigenvalue of the Hessian, $\lambda_{\max}$, and the low-pass filter (LPF) flatness measure (Bisla et al., 2022).

Flatness Metrics. We assess local geometry using Hessian eigenvalues and an LPF-inspired parameter sensitivity measure. For The largest eigenvalues of the estimated return Hessian characterize the extreme local curvatures (Sullivan et al., 2022). For our RL experiments, we use an LPF-inspired measure (Bisla et al., 2022) of the normalized absolute change in episodic return: $\widehat{\mathrm{LPF}}(\theta) = \mathbb{E}_{\delta\sim\mathcal{N}(0,\sigma^2)}\Big[L(\theta+\delta) - L(\theta)|/|L(\theta)|\Big]$. We use $M = 100$ independent perturbations with $\sigma = 0.01$ and reuse the same episode seeds across nominal and perturbed policies. Lower LPF values indicate lower relative sensitivity to parameter perturbations.

Reward Surface Visualization. We further visualize the reward landscape following the procedure of Sullivan et al. (2022). Figs. 16, 17 and 18 visualize the reward landscapes on HalfCheetah-v4, Hopper-v4, and Walker2d-v4 across five random seeds. SAM generally exhibits broader regions of relatively stable return, while mSAM also forms reasonably wide, relatively flat regions in most of the displayed landscapes. Compared with PPO and NAG, mSAM often shows more extended regions of high return, although some runs exhibit narrower, direction-dependent structures. These visualizations suggest that mSAM retains much of SAM's tendency to find locally flat solutions.

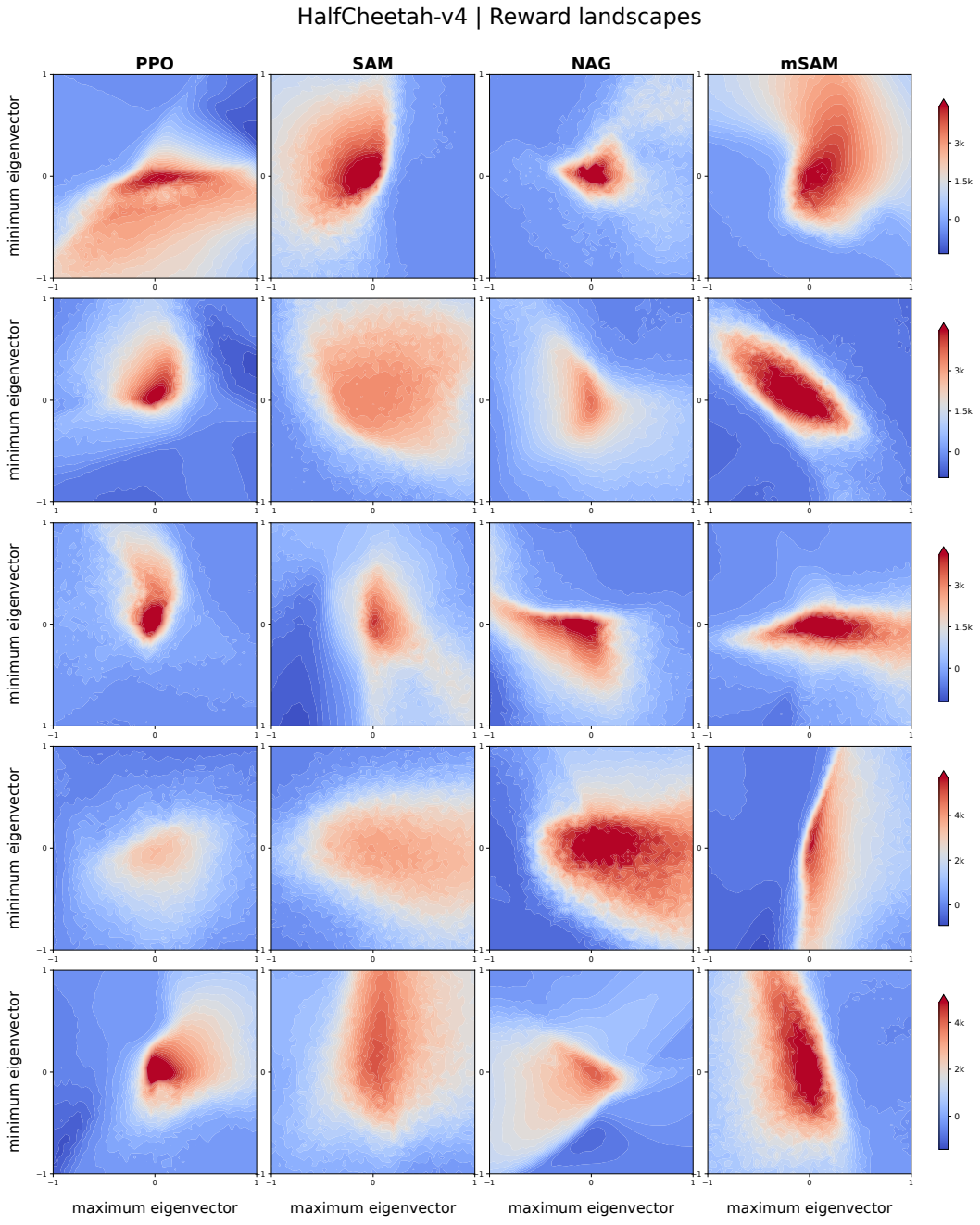


Figure 16: **Reward landscapes of PPO, SAM, NAG, and mSAM on HalfCheetah-v4.** Each row corresponds to a different random seed. Colors indicate the average episodic return.

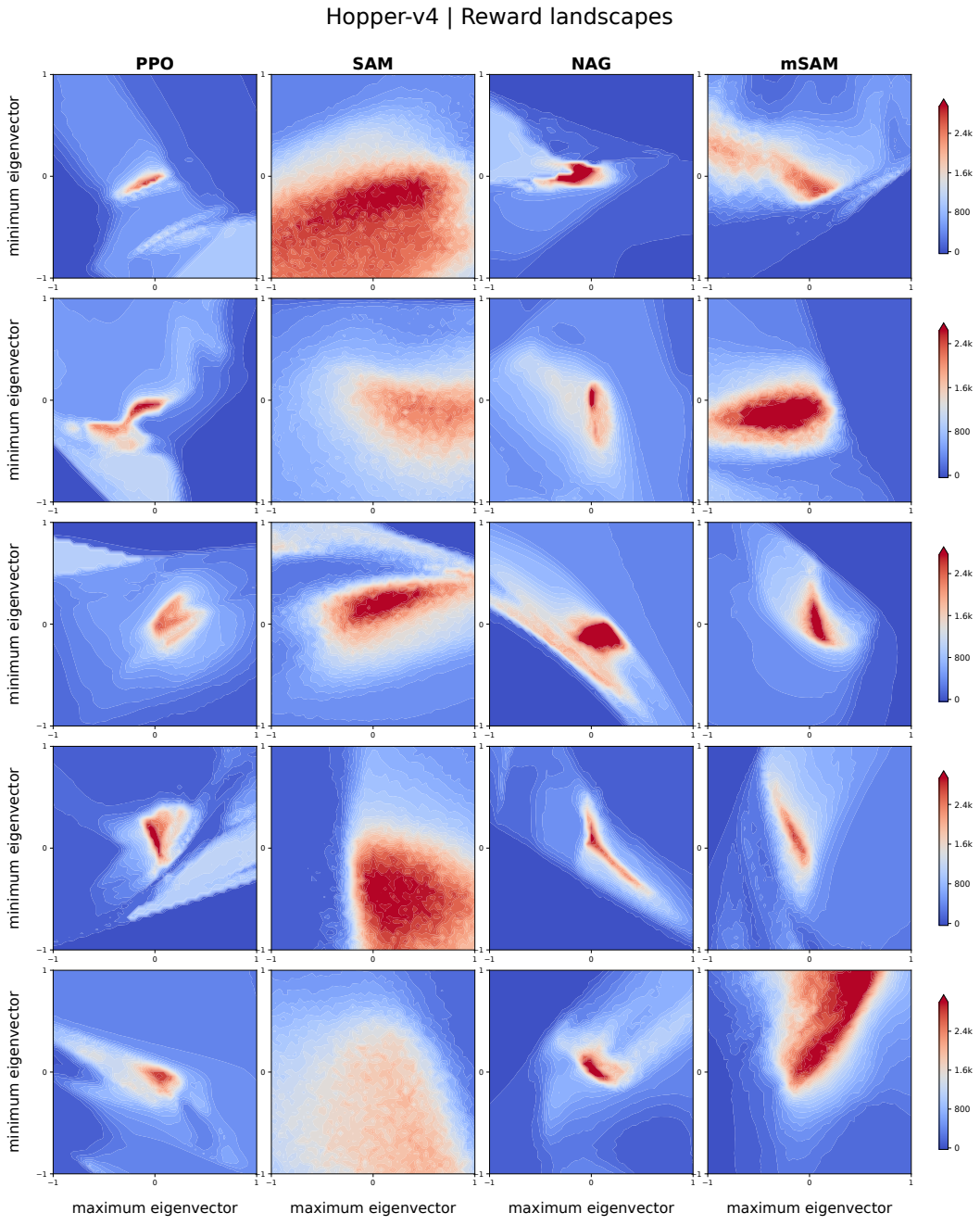


Figure 17: **Reward landscapes of PPO, SAM, NAG, and mSAM on Hopper-v4.** Each row corresponds to a different random seed. Colors indicate the average episodic return.

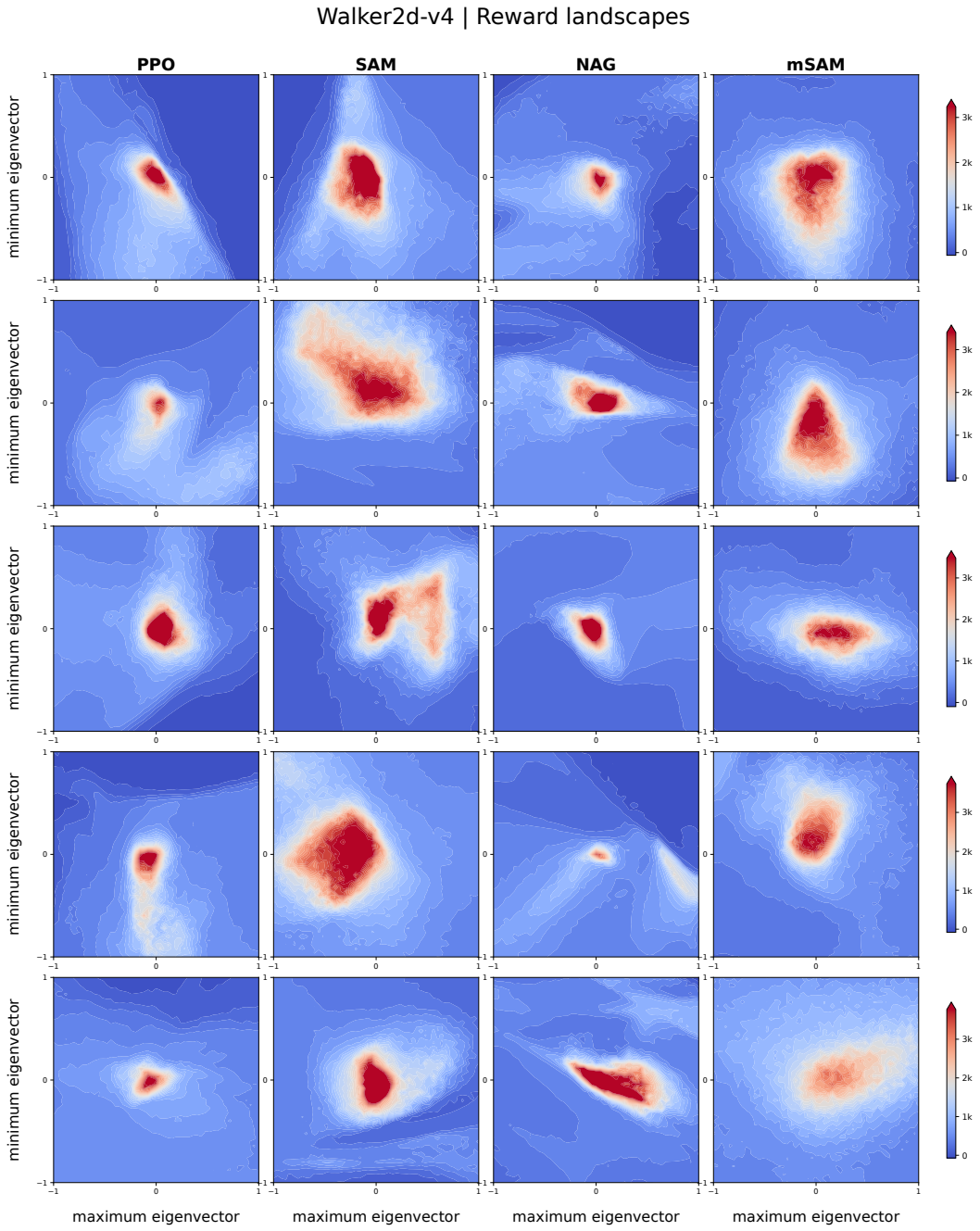


Figure 18: **Reward landscapes of PPO, SAM, NAG, and mSAM on Walker2d-v4.** Each row corresponds to a different random seed. Colors indicate the average episodic return.