BEYOND ENDPOINT FLATNESS: POLICY SCALE DYNAMICS SHAPE ROBUSTNESS IN SAM-PPO

Anonymous authors
Paper under double-blind review

ABSTRACT

Sharpness-Aware Minimization (SAM) improves PPO robustness to action noise, and prior work links this gain to flatter endpoint reward landscapes. We study how this robustness forms and what endpoint geometry can identify in diagonal-Gaussian PPO with learnable training scale, evaluating deterministic mean policies under external action noise. Across four MuJoCo tasks, replaying complete scale histories shows that action robustness largely follows the training-scale history, while changing the mean updater has smaller, task-dependent effects. Local analysis and routing interventions further support feedback from mean perturbations into the scale update, and explicit scale controls show that this pathway can be regulated without the full SAM update. With training scale fixed, Mean-SAM produces flatter relative reward landscapes without consistent action-noise gains. Equalizing the absolute parameter perturbation dose substantially weakens this flatness advantage, and further matching the induced action displacements reveals no consistent return-retention advantage across tasks. Together, these results identify training scale as a functional pathway behind SAM-PPO robustness and show that endpoint flatness alone cannot determine how that robustness was formed.

1 INTRODUCTION

Sharpness-Aware Minimization (SAM) seeks flatter minima by minimizing the worst-case loss within a parameter neighborhood (Foret et al., 2021). In supervised learning, mechanistic studies show that SAM's benefits arise from its optimization trajectory rather than endpoint sharpness alone (Andriushchenko & Flammarion, 2022; Agarwala & Dauphin, 2023; Wen et al., 2023; Vani et al., 2024). In reinforcement learning, Lee & Yoon (2025) link the action robustness of SAM-PPO to the flatness of its final reward landscape, connecting the two via a parameter-to-action mapping. Yet in Gaussian policies with learnable scale, SAM simultaneously alters both the mean and scale parameters during training. This work therefore uses systematic training interventions to separate these two roles: first identifying how training scale governs action robustness, and then evaluating what endpoint geometry actually captures once scale is controlled.

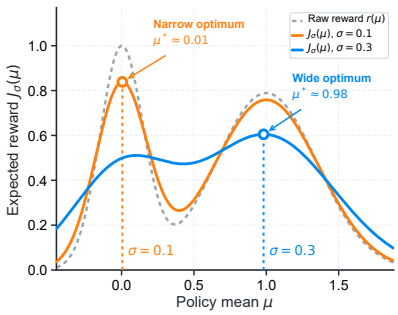

Figure 1: **Policy scale alters the objective.** In a one-step toy, changing the Gaussian scale shifts the expected-reward maximizer from a narrow to a broad reward peak.

The stochasticity of a Gaussian policy affects both the training process and the final policy (Papini et al., 2020; Montenegro et al., 2024; 2025). By changing the action-sampling distribution, policy scale changes the rewards and transitions sampled during training. A simple one-step example illustrates how this can shift the preferred policy mean between a narrow and a broad reward peak (Figure 1). Across four MuJoCo tasks, we find that SAM-PPO maintains a policy scale roughly four times larger than PPO late in training (Figure 2a), despite having no explicit scale control. This raises the question of whether SAM-PPO's robustness gains arise primarily through its altered training-scale history.

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.

To test this possibility, we replay the complete per-dimension scale histories of PPO and SAM-PPO as externally specified trajectories to policies trained from scratch under both PPO and Mean-SAM updaters. Here, Mean-SAM applies SAM only to the policy's mean parameters. Across the four tasks, action-noise robustness largely follows the replayed history. Once the history is fixed, switching the mean updater has a smaller effect that depends on the task and history. We then retrain PPO at several fixed scales. Moderately increasing the scale improves action-noise robustness, whereas an excessive scale harms nominal return. Together, these interventions show that training scale is a functional pathway that can be controlled independently, rather than merely a statistic accompanying SAM.

We next examine one local source of SAM's larger scale. A local first-order analysis suggests that mean perturbations in SAM generate mean-to-scale feedback that opposes scale contraction. Routing experiments support this account: sending this induced feedback to the scale block maintains a larger training scale and yields stronger action-noise robustness. Entropy regularization and mechanism-inspired explicit controls further show that this scale pathway can be regulated without SAM.

Finally, we fix the policy scale throughout training and compare standard PPO updates with SAM updates applied only to the policy mean (Mean-SAM). On standard relatively normalized grids, Mean-SAM yields higher return retention without consistent action-noise gains. Because Mean-SAM learns smaller weight norms, filter normalization injects smaller absolute parameter perturbations and produces smaller action displacements. After equalizing the absolute parameter perturbation dose, the four-task mean flatness gap falls to near zero. It turns negative in three tasks and stays positive in Ant. After further matching the induced action displacements, Mean-SAM shows no consistent return-retention advantage across tasks. These results show that interpreting endpoint flatness requires examining both the parameter perturbation dose and the action changes it induces.

We use three complementary interventions to separate the roles of training scale and the mean updater: scale-history replay, feedback routing, and fixed-scale controls. Across these interventions, scale history has the larger and more consistent effect on action-noise robustness. Changing the mean updater has smaller, task-dependent effects on this robustness, while altering parameter-space return retention, absolute perturbation dose, and realized action displacement. Endpoint geometry therefore conflates distinct mechanisms, and cannot alone identify how action robustness forms during training.

2 PRELIMINARIES

2.1 DIAGONAL GAUSSIAN POLICY AND POLICY SCALE

We consider a diagonal Gaussian policy with a state-dependent mean and a state-independent, per-dimension policy scale:

$$\pi(a \mid s;\theta) = \mathcal{N}\big(\mu(s;\psi), \mathrm{diag}(\sigma^2)\big), \qquad \theta = (\psi, \phi), \qquad \sigma_j = e^{\phi_j}.$$

Here ψ parameterizes the action mean, while ϕ parameterizes the log-scale. During training, the policy samples actions from this distribution and clips them to the environment bounds; the mean-network output is bounded by $\tanh$. At evaluation time, we use the deterministic mean action $a = \mu(s;\psi)$ and do not sample from the learned σ.

We summarize the overall scale by its geometric mean. For the pre-clipping Gaussian policy, its entropy satisfies

$$\sigma_{\mathrm{GM}} = \exp\left(\frac{1}{n}\sum_{j=1}^{n}\phi_j\right), \qquad \mathcal{H}(\pi) = \frac{n}{2}\log(2\pi e) + \sum_{j=1}^{n}\phi_j = \frac{n}{2}\log(2\pi e) + n\log\sigma_{\mathrm{GM}}.$$

Thus, entropy regularization can directly adjust the log-scale. The scale-history interventions below always operate on the full per-dimension vector σ, while σ_{GM} only summarizes its overall level. Thus, any effect of the training scale on the final deterministic policy must arise through training.

2.2 PPO AND ACTOR-ONLY SAM

Let $L(\theta)$ denote the standard clipped PPO actor loss (Schulman et al., 2017); its full form and training configuration are given in Appendix B.1. We write

$$g = (g_\psi, g_\phi) = \nabla_\theta L, \qquad G = \|g\|_2.$$

SAM perturbs parameters along $\delta_{\mathrm{SAM}} = \rho g/G$, recomputes the gradient at the perturbed point, restores the original parameters, and performs the outer update (Foret et al., 2021). We apply SAM only to the actor; the critic is not perturbed, and the mean and scale blocks are normalized jointly with a shared perturbation radius ρ.

Let $u_t = a_t - \mu(s_t;\psi)$ denote the action residual. For a fixed state and sampled action, the Gaussian policy satisfies

$$\frac{\partial \log \pi(a_t \mid s_t;\theta)}{\partial \mu_j} = \frac{u_{t,j}}{\sigma_j^2}, \qquad \frac{\partial \log \pi(a_t \mid s_t;\theta)}{\partial \phi_j} = \frac{u_{t,j}^2}{\sigma_j^2} - 1.$$

Even though the mean network does not explicitly depend on ϕ, its policy gradient depends on policy scale, a dependence that Section 4 traces into SAM's scale update.

3 DOES POLICY SCALE HELP FORM ROBUSTNESS?

We conduct our core experiments on HalfCheetah-v4, Hopper-v4, Walker2d-v4, and Ant-v4 (Todorov et al., 2012; Towers et al., 2024). Unless stated otherwise, each setting is trained for 3 million environment steps across 15 random seeds, and all evaluations use 50 episodes per condition at the final checkpoint. The actor and critic each use two-layer tanh networks with 64 units, and SAM uses $\rho = 0.008$. The complete configuration is given in Appendix B.1.

We evaluate action-noise robustness by adding external Gaussian noise to deterministic mean actions, following Lee & Yoon (2025). The noise scale τ spans 11 equally spaced levels in $[0, 0.5]$. Denoting the average return at noise level τ by $J_{\mathrm{noise}}(\tau)$, action AUC and its normalized counterpart are defined as

$$\mathrm{AUC} = \int_0^{0.5} J_{\mathrm{noise}}(\tau)\,\mathrm{d}\tau, \qquad \mathrm{nAUC} = \frac{\mathrm{AUC}}{0.5\,J_{\mathrm{noise}}(0)}. \tag{1}$$

Integrals are computed using the trapezoidal rule. We compute nAUC per seed before averaging, measuring return retention relative to the zero-noise baseline. We also report nominal returns from an independent evaluation. Appendix B.1 provides full evaluation and statistical protocols.

Table 1: **Native PPO and SAM-PPO comparison.** Values are mean ± SE. The shaded column highlights policy scale; parentheses report nAUC as a percentage. Bold marks larger policy scale and higher return and AUC within each environment, not statistical significance.

Environment	Method	Policy Scale	Nominal Return	Action AUC (nAUC)
HalfCheetah	PPO	0.041 ± 0.001	5445 ± 289	1017 ± 67 (38.5%)
	SAM-PPO	$\mathbf{0.156 \pm 0.007}$	$\mathbf{6843 \pm 334}$	$\mathbf{1967 \pm 131}$ (57.7%)
Hopper	PPO	0.056 ± 0.002	$\mathbf{3125 \pm 146}$	518 ± 39 (33.5%)
	SAM-PPO	$\mathbf{0.223 \pm 0.005}$	3078 ± 241	$\mathbf{795 \pm 37}$ (56.8%)
Walker2d	PPO	0.070 ± 0.003	4574 ± 259	876 ± 60 (39.0%)
	SAM-PPO	$\mathbf{0.292 \pm 0.010}$	$\mathbf{5168 \pm 80}$	$\mathbf{1860 \pm 80}$ (71.5%)
Ant	PPO	0.039 ± 0.001	3993 ± 214	700 ± 34 (34.5%)
	SAM-PPO	$\mathbf{0.168 \pm 0.002}$	$\mathbf{5644 \pm 53}$	$\mathbf{1204 \pm 12}$ (42.8%)

3.1 SAM CHANGES BOTH SCALE AND PERFORMANCE

Table 1 compares native PPO and SAM-PPO in policy scale, nominal return, and action-noise AUC. Across all four tasks, the final scale of SAM-PPO is approximately 3.8–4.3 times that of PPO, and

its action-noise AUC is also consistently higher. Figure 2(a) shows the complete trajectories: this scale gap emerges early and persists throughout training. Nominal return improves on three tasks, while the two methods are close on Hopper. This establishes the phenomenon we aim to explain: SAM's action-noise robustness gain is accompanied by a systematic change in training scale.

This scale gap also changes the reading of endpoint sharpness. The curvature of the PPO loss in the mean parameters depends on σ through the log-likelihood. At their own training scales, the largest mean-block curvature of SAM-PPO is less than one tenth of that of PPO. When both policies use the same common σ, the median ratio becomes 0.6–2.0 (Appendix C.3). Loss-based sharpness gaps therefore partly reflect differences in training scale. We next intervene on the training scale directly.

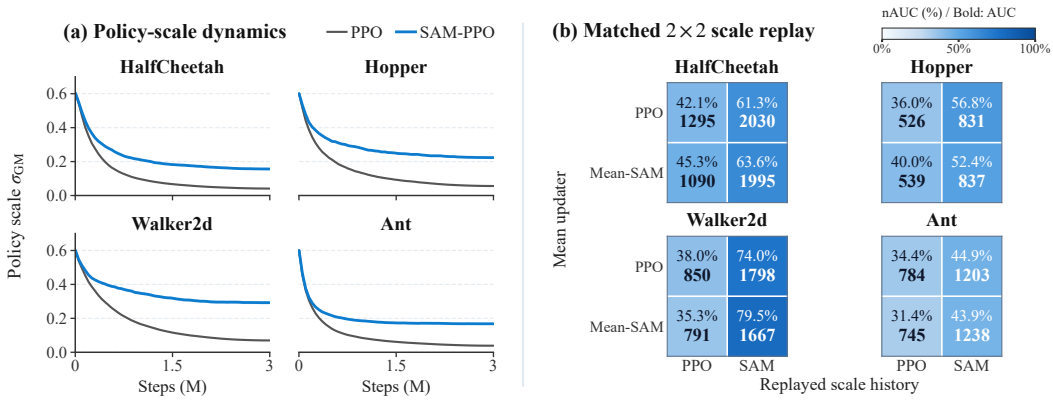

Figure 2: **Scale dynamics and matched replay.** (a) Native PPO and SAM-PPO policy-scale trajectories (σ_{GM}); curves and shaded bands show the mean ± 1 SE over 15 seeds. (b) Receivers trained from scratch with either mean updater (rows) and either replayed scale history (columns). Cell colors and percentages show mean nAUC; bold numbers show mean action-noise AUC over 15 seeds.

3.2 Swapping Scale Histories

If scale history carries SAM's action-robustness benefit, replaying SAM's history should improve action-noise AUC under either mean updater. For each seed, we record the complete per-dimension scale histories of PPO and SAM-PPO and replay them as externally specified trajectories to receivers trained from scratch. Each receiver interacts with the environment on its own with scale prescribed by the replayed history; its mean parameters are updated by either standard PPO or SAM applied only to the mean network (Mean-SAM). This yields four combinations of two mean updaters and two scale histories, including two self-replay controls.

Figure 2(b) shows that replacing the PPO scale history with the SAM-PPO history increases action-noise AUC under both mean updaters in all four tasks. Switching the mean updater while holding the history fixed produces smaller AUC changes, whose direction depends on the task and history. In Ant, for example, the estimated effect of switching from PPO to Mean-SAM is negative under the PPO history and positive under the SAM-PPO history. Nominal returns and complete paired statistics are given in Appendix C.1. This intervention changes the scale level, timing, and per-dimension structure together. It therefore shows that the full history can carry differences in action-noise robustness, but does not identify which aspect is responsible. We next isolate the scale level.

3.3 Directly Specifying the Scale Level

To test whether changing the scale level alone can improve action-noise robustness, we next remove temporal variation and use a common, constant scale across action dimensions. Before training, we fix the scale of each PPO action dimension to 0.05, 0.10, 0.20, 0.40, or 0.80. This experiment uses neither the SAM mean update nor a learned scale schedule. As the scale increases from 0.05 to 0.20, action-noise AUC increases across all four tasks, consistent with the replay findings (Figure 3); increasing the scale to 0.40 raises AUC further. The scale that maximizes nominal return depends

on the task. When the scale reaches 0.80, nominal return drops substantially across all four tasks. Complete dose-response data are reported in Appendix C.2.

Together, the two interventions show that training scale is a functional variable that can be controlled independently, and that PPO without the SAM mean update can obtain action-noise gains through scale alone.

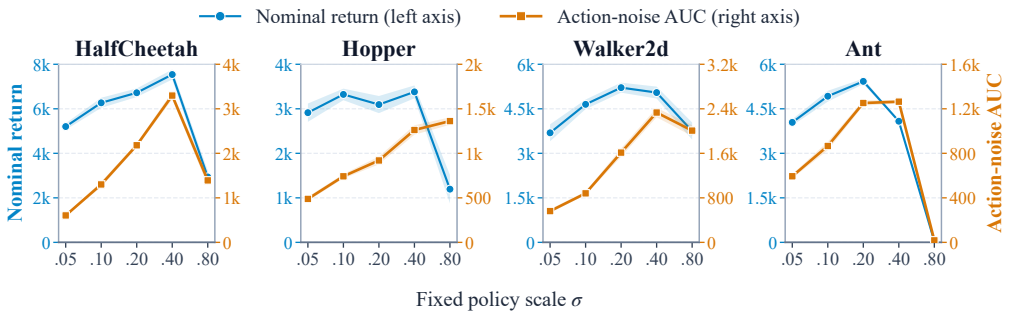

Figure 3: **Fixed-scale dose response.** Nominal return (blue, left axis) and action-noise AUC (amber, right axis) across fixed policy scales $\sigma \in \{0.05, 0.10, 0.20, 0.40, 0.80\}$. Curves and shaded bands indicate mean $\pm$ SE across 15 seeds per condition.

4 Where Does Scale Change Come From, and Can It Be Controlled Explicitly?

4.1 Feedback from the Mean Gradient to the Scale

On a fixed minibatch, we keep the active PPO clipping branch unchanged and assume that the local loss is smooth with $G > 0$. The first-order additional SAM gradient is $\rho \nabla G$, consistent with gradient-norm views of sharpness-aware updates (Andriushchenko & Flammarion, 2022; Zhao et al., 2022). We denote the corresponding additional descent feedback on the log-scale coordinates by

$$F = -\rho \nabla_\phi G = F_{\text{cross}} + F_{\text{ratio}} + F_\phi.$$

Here, F_{cross} comes from the explicit inverse-variance factors in the mean gradient, F_{ratio} from the scale dependence of its probability-ratio weights, and F_ϕ from the scale-gradient contribution to G. The following proposition characterizes F_{cross}.

Proposition 1 (Local mean-to-scale feedback)**.** *For a diagonal Gaussian policy with state-independent scale, assume that the mean network does not explicitly depend on ϕ. If $g_{\psi,j}$ denotes the contribution of action coordinate j to the mean gradient and $\sum_j g_{\psi,j} = g_\psi$, then*

$$F_{\text{cross},j} = \frac{2\rho}{G} \langle g_\psi, g_{\psi,j} \rangle, \qquad \sum_j F_{\text{cross},j} = \frac{2\rho}{G} \|g_\psi\|_2^2 \geq 0.$$

The nonnegative coordinate sum identifies an entropy-increasing component of the local descent feedback whenever the mean gradient is nonzero. The next section tests the training role of the mean-to-scale channel by altering where mean-perturbation feedback is received. Appendix A.1 provides the full decomposition and proof.

4.2 Changing Where the Feedback Is Received

The Gaussian gradient identities in Section 2.2 show that perturbing the mean changes the action residual entering the scale gradient. To test this mean-to-scale channel during training, we perturb only the mean parameters and alter where the additional gradient is received. Both routes build on standard PPO gradients: $\psi \to \phi$ adds the mean-induced perturbation gradient only to the scale block, whereas $\psi \to \psi$ adds it only to the mean block. Both routes match the source perturbation

norm and rescale the receiving gradient to its clean clipped norm. Thus, $\psi \to \phi$ tests the training effect of mean-induced scale feedback under this norm control; the local analysis identifies F_{cross} as one source of that feedback. Full routing details are given in Appendix B.2.

Compared with $\psi \to \psi$, the $\psi \to \phi$ route produces a larger final scale in all four tasks and higher point estimates for action-noise AUC. For example, in Hopper, $\psi \to \phi$ reaches a scale of 0.194 and an AUC of 789, compared with 0.069 and 537 for $\psi \to \psi$. The AUC of $\psi \to \phi$ is close to native SAM in Hopper and Ant, but remains clearly lower in HalfCheetah and Walker2d. The complete four-route results are given in Appendix D.1. These results support a role for mean-to-scale feedback in producing the scale phenotype in full training.

4.3 Explicit Scale Control

We next ask whether explicit scale regulation can recover substantial action-noise gains without the full SAM update. Unlike the fixed-scale experiments, these controls keep scale learnable and counteract its contraction through simple regularization. Entropy regularization provides a general control (Ahmed et al., 2019; Geist et al., 2019), with β selected to minimize the trajectory RMSE relative to SAM rather than to maximize robustness. Motivated by the inverse-variance dependence of the mean gradient in Section 4.1, regularized feedback descent (RFD) instead adds a scale penalty to the actor loss:

$$L_{\text{RFD}} = L_{\text{PPO}} + \frac{\lambda_k}{2} \sum_j \sigma_j^{-2}, \tag{2}$$

where L_{PPO} is the PPO actor loss from Section 2.2. The coefficient λ_k is fixed throughout update k using gradient statistics from the prior update. Its additional descent feedback on log-scale coordinate j is $\lambda_k \sigma_j^{-2}$, giving smaller scales a stronger upward adjustment. Appendix B.2 details coefficient construction and entropy calibration.

Both controls have higher action-noise AUC point estimates than PPO across the four tasks (Table 2). RFD also exceeds native SAM in AUC point estimates but has lower nominal return in three tasks, while constant entropy regularization selected by scale-trajectory RMSE does not match SAM on every task. Thus, the scale pathway can be controlled explicitly without the full SAM update, but different controls lead to different trade-offs between nominal performance and action-noise robustness.

Table 2: **Explicit scale controls.** Values are mean ± SE over 15 seeds. Bold marks the highest value for each metric within each environment, not statistical significance.

Environment	Metric	PPO	SAM-PPO	Entropy-PPO	RFD-PPO
HalfCheetah	Action AUC	1017 ± 67	1967 ± 131	1921 ± 94	**2205 ± 136**
	Nominal Return	5445 ± 289	**6843 ± 334**	6678 ± 385	6384 ± 427
Hopper	Action AUC	518 ± 39	795 ± 37	759 ± 43	**1133 ± 43**
	Nominal Return	3125 ± 146	3078 ± 241	3322 ± 144	**3338 ± 167**
Walker2d	Action AUC	876 ± 60	1860 ± 80	1365 ± 138	**2033 ± 90**
	Nominal Return	4574 ± 259	**5168 ± 80**	4686 ± 225	4586 ± 274
Ant	Action AUC	700 ± 34	1204 ± 12	1079 ± 28	**1286 ± 39**
	Nominal Return	3993 ± 214	**5644 ± 53**	4902 ± 154	4939 ± 139

5 What Does Endpoint Flatness Reflect?

The preceding experiments demonstrate that the training scale maintained by SAM-PPO shapes the policy's action-noise robustness. Directly comparing the endpoint geometry of native PPO and SAM-PPO therefore conflates the learned scale pathway with the effect of the mean updater. This raises a straightforward question: if the training scale is held fixed throughout optimization, does the mean updater by itself produce a flatter endpoint? And if so, does this parameter-space flatness

translate into action-noise robustness? We address this question by fixing the training scale during training, jointly measuring parameter-space return retention and realized action displacements.

5.1 Measuring Return and Action Changes Together

Around the final mean parameters, we draw two random directions, apply relative normalization to the mean-network weights row by row without perturbing biases, and evaluate returns on an 11×11 grid over $[-0.20, 0.20]^2$. This follows the relative direction-normalization procedure commonly used in two-dimensional landscape visualizations (Li et al., 2018; Sullivan et al., 2022; Lee & Yoon, 2025). Complete details on direction orthogonalization, row normalization, and random streams are provided in Appendix B.3.

At each grid point, we average returns over 50 complete evaluation episodes under the perturbed policy. A_{80} is the fraction of points retaining at least 80% of the unperturbed policy's mean return. We measure action displacement separately on 500 shared reference states collected from the native PPO policy with the same training seed:

$$d_\psi(q) = \frac{1}{N} \sum_{s \in \mathcal{S}} \left\| \mu_{\psi + \delta\psi(q)}(s) - \mu_\psi(s) \right\|_2, \qquad \delta\psi(q) = \alpha_1 d_1 + \alpha_2 d_2.$$

In short, each parameter perturbation yields both a perturbed return and an action displacement; across the grid, A_{80} summarizes overall return retention, $\bar{d}$ summarizes average action displacement, and D_{rms} denotes the root-mean-square Euclidean norm of the parameter perturbations over the full grid. Across models, evaluations share the same normalization rules and paired random direction seeds. Below, we compare behavioral performance, parameter flatness, parameter dose, and realized action displacement side by side in a controlled fixed-scale experiment.

5.2 Isolating the Mean Updater at Fixed Training Scale

To isolate the effect of the mean updater, we fix the policy scale of both PPO and Mean-SAM to 0.20 throughout training, varying only the mean update rule. Standard PPO uses its regular mean update, whereas Mean-SAM applies SAM perturbations and updates solely to the mean network while holding the scale parameters frozen.

Table 3: **Fixed-scale mean-updater comparison ($\sigma = 0.20$).** Values are mean $\pm$ SE over 15 seeds; parentheses report nAUC as a percentage. Bold marks higher behavioral performance and A_{80}, but lower displacement and parameter dose, within each environment.

Environment	Updater	Behavioral performance		Parameter-space diagnostics		
		Nominal Return	Action AUC (nAUC)	Flatness A_{80}	Displacement $\bar{d}$	Parameter dose D_{rms}
HalfCheetah	PPO	**6721 ± 194**	**2181 ± 49** (65.7%)	0.461 ± 0.031	0.356 ± 0.010	2.981 ± 0.016
	Mean-SAM	6200 ± 156	1981 ± 51 (63.1%)	**0.552 ± 0.036**	**0.285 ± 0.012**	**2.486 ± 0.035**
Hopper	PPO	3095 ± 192	920 ± 44 (61.9%)	0.447 ± 0.039	0.162 ± 0.012	2.649 ± 0.019
	Mean-SAM	**3342 ± 137**	**946 ± 34** (57.6%)	**0.528 ± 0.045**	**0.145 ± 0.007**	**1.963 ± 0.018**
Walker2d	PPO	**5213 ± 169**	**1613 ± 74** (61.9%)	0.349 ± 0.032	0.318 ± 0.014	2.669 ± 0.008
	Mean-SAM	4900 ± 210	1554 ± 82 (63.2%)	**0.449 ± 0.040**	**0.285 ± 0.028**	**2.078 ± 0.010**
Ant	PPO	**5424 ± 94**	1252 ± 24 (46.3%)	0.718 ± 0.034	0.344 ± 0.009	2.767 ± 0.003
	Mean-SAM	5383 ± 51	**1275 ± 14** (47.0%)	**0.970 ± 0.017**	**0.223 ± 0.012**	**2.058 ± 0.009**

Table 3 shows higher A_{80} point estimates for Mean-SAM in all four tasks, but no consistent action-noise AUC gain. In Ant, A_{80} increases from 0.718 to 0.970, yet action-noise AUC remains essentially unchanged (1252 versus 1275); in HalfCheetah, A_{80} likewise increases, but action-noise AUC decreases from 2181 to 1981.

Under relative normalization, the perturbation magnitude scales with the weight norm. This agrees with the known sensitivity of flatness to parameter rescaling (Dinh et al., 2017; Lyu et al., 2022). On

relatively normalized grids, Mean-SAM receives an absolute parameter RMS dose only 0.74–0.83 times that of PPO at the same training scale, and its mean action displacements are also lower. We therefore test whether the flatness comparison depends on parameter dose. We retain each model's perturbation directions and rescale its grid radius so that its full-grid D_{rms} matches that of PPO with the same training seed. The equally weighted mean A_{80} difference (Mean-SAM minus PPO) across the four tasks decreases from approximately +0.131 to +0.004; Ant retains a positive difference (+0.133), while the differences in the other three tasks become negative (Appendix E.2). Thus, the flatness advantage under relative normalization is sensitive to the absolute parameter perturbation dose. We next compare return retention at grid points with similar action displacements.

5.3 Return Retention at Similar Action Displacements

We compare PPO and Mean-SAM trained with scale fixed at $\sigma = 0.20$. On the common parameter RMS dose grids, we match noncentral points one to one without replacement within a 5% IQR tolerance of mean action displacement for each environment and training seed, and compare the difference in the fraction of matched points retaining at least 80% of their unperturbed returns (Table 4; matching details in Appendix E.2).

Table 4: **Return retention after action-displacement matching.** Retention-fraction differences are Mean-SAM minus PPO, reported as paired means ± SE. Matched-pair counts are averaged across seeds, with 120 candidate points per model.

Measure	HalfCheetah	Hopper	Walker2d	Ant
Retention-fraction difference	-0.032 ± 0.047	$+0.017 \pm 0.060$	-0.020 ± 0.079	$+0.054 \pm 0.049$
Matched pairs	94.9	79.4	76.5	84.1

After matching action displacement, Mean-SAM shows no consistent return-retention advantage across tasks: the retention-fraction differences are negative in HalfCheetah and Walker2d and positive in Hopper and Ant. Changing the matching tolerance or using displacement-bin weighting yields the same conclusion (Appendix E.2). Thus, higher flatness on relative grids does not translate into a consistent return-retention advantage at similar action displacements within the region of overlap.

5.4 A SAM-Free L2-Regularization Control

To test whether the separation between parameter flatness and action-noise robustness is specific to SAM, we apply L2 regularization exclusively to the mean-network weights of standard PPO with fixed $\sigma = 0.20$. We evaluate three pre-specified regularization strengths $\lambda \in \{3 \times 10^{-5}, 10^{-4}, 3 \times 10^{-4}\}$ on HalfCheetah and Ant, yielding six environment–strength comparisons.

On relatively normalized grids, all six L_2-regularization comparisons have lower absolute parameter doses and lower mean action-displacement estimates than the unregularized baseline. Their A_{80} estimates are higher, but action-noise AUC does not improve consistently. At a common parameter RMS dose, five of the six A_{80} difference estimates become negative, while the weakest regularization setting in Ant yields a near-zero difference. Subsequent action-displacement matching reveals no consistent return-retention advantage, with a negative difference remaining for the strongest regularization setting in Ant (Appendix E.3). Thus, the flatness advantage of L_2 regularization on relative grids also depends on the evaluation perturbation dose.

Together, the dose controls and displacement matching show that interpreting endpoint flatness requires accounting for both the parameter perturbation magnitude and the action changes it induces.

6 Related Work

Flatness and reward landscapes. Flatness is commonly used to analyze optimization and generalization (Keskar et al., 2017), but its numerical value depends on the parameterization and perturbation convention (Dinh et al., 2017; Lyu et al., 2022). In scale-invariant networks, changing weight

norms can alter reported sharpness without changing the represented function (Lyu et al., 2022). Li et al. (2018) introduced normalized two-dimensional landscape visualizations, and Sullivan et al. (2022) applied them to reward surfaces in reinforcement learning. Lee & Yoon (2025) further linked parameter-space flatness to action robustness. We extend this line by separating training-scale and mean-updater effects and by jointly examining parameter dose, action displacement, and action-noise robustness.

SAM beyond endpoint sharpness. Foret et al. (2021) introduced SAM, and subsequent work modified its perturbation scale or optimization objective (Kwon et al., 2021; Zhuang et al., 2022; Singh et al., 2025; Zhao et al., 2022). Mechanistic studies show that SAM's effects cannot be summarized by endpoint sharpness alone (Andriushchenko & Flammarion, 2022), and that the training trajectory itself also matters (Agarwala & Dauphin, 2023; Vani et al., 2024). The parameter blocks receiving the adversarial perturbation can also matter: perturbing only affine normalization parameters can outperform perturbing all parameters in some architectures (Mueller et al., 2023). This blockwise perspective is related to our routing and fixed-scale interventions, although our policy blocks are Gaussian mean and scale parameters rather than normalization-layer parameters. In reinforcement learning, sharpness-aware updates have been applied to action robustness, safe exploration, and offline generalization (Lee & Yoon, 2025; Mani et al., 2026; Wang et al., 2026).

Policy stochasticity as a training variable. The stochasticity of a Gaussian policy affects both learning stability and the final solution (Ahmed et al., 2019; Papini et al., 2020; Montenegro et al., 2024; 2025). Meanwhile, maximum-entropy objectives explicitly control stochasticity through another route (Haarnoja et al., 2018; Eysenbach & Levine, 2022), but excessive entropy can also hinder precise control (Zhang et al., 2025). We study policy scale as a training variable in SAM-PPO.

Action perturbations, regularization, and parameter sensitivity. Action-robust RL formalizes performance under execution-time action perturbations (Tessler et al., 2019), while parameter-space noise and optimization-time policy-mean perturbations provide complementary routes for perturbing policies during learning (Plappert et al., 2018; Rahman et al., 2025). We use external action noise as a direct behavioral evaluation to test robustness across training pathways. Network regularization can substantially alter policy optimization outcomes (Liu et al., 2021), while spectral-norm products provide structural bounds on perturbation sensitivity in deep networks (Bartlett et al., 2017; Neyshabur et al., 2018).

7 Discussion and Conclusion

Our interventions identify training-scale history as a major pathway through which SAM-PPO improves action-noise robustness in the settings studied. With scale fixed, changing the mean updater still alters endpoint flatness, absolute perturbation dose, and realized action displacement, without consistent action-noise gains. Training scale helps shape action robustness. Endpoint geometry, in turn, depends on the measurement convention: loss-based sharpness varies with the training scale, and return flatness varies with the magnitude of the perturbations used to probe it.

This result suggests a guideline for comparing stochastic policy optimizers: in addition to final returns and reward landscapes, researchers should inspect the training scale and, when needed, fix its history before comparing update rules. The entropy and RFD controls show that explicit scale regulation can produce strong action-noise robustness, though nominal return must still be evaluated separately. We do not yet disentangle the distinct roles of the overall scale level, its temporal schedule, and its per-dimension structure.

Our analysis focuses on actor-only SAM with diagonal Gaussian policies across four continuous-control benchmarks. We measure flatness on a finite set of random two-dimensional slices. Action-displacement matching uses mean action changes on shared reference states and restricts the comparison to points that can be matched. The main comparison reports absolute parameter dose and realized action displacement; structural weight norms remain an accompanying diagnostic rather than a standalone explanation of robustness. Additional evaluations on BipedalWalker-v3 and LunarLanderContinuous-v3 examine this scope in Appendix F.1.

REFERENCES

Atish Agarwala and Yann Dauphin. SAM operates far from home: eigenvalue regularization as a dynamical phenomenon. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 152–168, 2023. doi: 10.48550/arXiv.2302.08692. URL https://proceedings.mlr.press/v202/agarwala23a.html.

Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In *International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 151–160, 2019. doi: 10.48550/arXiv.1811.11214. URL https://proceedings.mlr.press/v97/ahmed19a.html.

Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In *International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 639–668, 2022. doi: 10.48550/arXiv.2206.06232. URL https://proceedings.mlr.press/v162/andriushchenko22a.html.

Peter L. Bartlett, Dylan J. Foster, and Matus J. Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/b22b257ad0519d4500539da3c8bcf4dd-Abstract.html.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1019–1028. PMLR, 2017. URL https://proceedings.mlr.press/v70/dinh17b.html.

Benjamin Eysenbach and Sergey Levine. Maximum entropy RL (provably) solves some robust RL problems. In *International Conference on Learning Representations (ICLR)*, 2022. doi: 10.48550/arXiv.2103.06257. URL https://openreview.net/forum?id=PtSAD3caaA2.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations (ICLR)*, 2021. doi: 10.48550/arXiv.2010.01412. URL https://arxiv.org/abs/2010.01412.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2160–2169, 2019. URL https://proceedings.mlr.press/v97/geist19a.html.

Shangding Gu, Laixi Shi, Muning Wen, Ming Jin, Eric Mazumdar, Yuejie Chi, Adam Wierman, and Costas Spanos. Robust gymnasium: A unified modular benchmark for robust reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=wIm22TJvS7.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870, 2018. doi: 10.48550/arXiv.1801.01290. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

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 (ICLR)*, 2017. doi: 10.48550/arXiv.1609.04836. URL https://arxiv.org/abs/1609.04836.

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 (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5905–5914, 2021. doi: 10.48550/arXiv.2102.11600. URL https://proceedings.mlr.press/v139/kwon21b.html.

Hyun Kyu Lee and Sung Whan Yoon. Flat reward in policy parameter space implies robust reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=4OaO3GjP7k.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. doi: 10.48550/arXiv.1712.09913. URL https://arxiv.org/abs/1712.09913.

Zhuang Liu, Xuanlin Li, Bingyi Kang, and Trevor Darrell. Regularization matters in policy optimization: An empirical study on continuous control. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=yr1mzrH3IC.

Kaifeng Lyu, Zhiyuan Li, and Sanjeev Arora. Understanding the generalization benefit of normalization layers: Sharpness reduction, 2022. URL https://arxiv.org/abs/2206.07085.

Kaustubh Mani, Yann Pequignot, Vincent Mai, and Liam Paull. SHAPO: Sharpness-aware policy optimization for safe exploration. In *International Conference on Learning Representations (ICLR)*, 2026. doi: 10.48550/arXiv.2606.10228. URL https://arxiv.org/abs/2606.10228.

Alessandro Montenegro, Marco Mussi, Alberto Maria Metelli, and Matteo Papini. Learning optimal deterministic policies with stochastic policy gradients. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 36160–36211. PMLR, 2024. URL https://proceedings.mlr.press/v235/montenegro24a.html.

Alessandro Montenegro, Marco Mussi, Matteo Papini, and Alberto Maria Metelli. Convergence analysis of policy gradient methods with dynamic stochasticity. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 44652–44698. PMLR, 2025. URL https://proceedings.mlr.press/v267/montenegro25a.html.

Maximilian Mueller, Tiffany Vlaar, David Rolnick, and Matthias Hein. Normalization layers are all that sharpness-aware minimization needs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. doi: 10.48550/arXiv.2306.04226. URL https://arxiv.org/abs/2306.04226.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://openreview.net/forum?id=Skz_WfbCZ.

Matteo Papini, Andrea Battistello, and Marcello Restelli. Balancing learning speed and stability in policy gradient via adaptive exploration. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1188–1199, 2020. URL https://proceedings.mlr.press/v108/papini20a.html.

Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. In *International Conference on Learning Representations (ICLR)*, 2018. doi: 10.48550/arXiv.1706.01905. URL https://arxiv.org/abs/1706.01905.

Md Masudur Rahman, Juan P. Wachs, and Yexiang Xue. Robust policy gradient optimization through action parameter perturbation in reinforcement learning. In *Proceedings of the Aligning Reinforcement Learning Experimentalists and Theorists Workshop (ARLET@NeurIPS 2025) at the Thirty-Ninth Annual Conference on Neural Information Processing Systems (NeurIPS 2025)*, 2025. URL https://openreview.net/forum?id=B6N67euqyv.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Sidak Pal Singh, Hossein Mobahi, Atish Agarwala, and Yann Dauphin. Avoiding spurious sharpness minimization broadens applicability of SAM. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55702–55719. PMLR, 2025. URL https://proceedings.mlr.press/v267/singh25b.html.

Ryan Sullivan, Jordan K. Terry, Benjamin Black, and John P. Dickerson. Cliff diving: Exploring reward surfaces in reinforcement learning environments. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20744–20776. PMLR, 2022. URL https://proceedings.mlr.press/v162/sullivan22a.html.

Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6215–6224. PMLR, 2019. URL https://proceedings.mlr.press/v97/tessler19a.html.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109. URL https://doi.org/10.1109/IROS.2012.6386109.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments, 2024. URL https://arxiv.org/abs/2407.17032.

Ankit Vani, Frederick Tung, Gabriel L. Oliveira, and Hossein Sharifi-Noghabi. Forget sharpness: Perturbed forgetting of model biases within SAM dynamics. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 49122–49136, 2024. doi: 10.48550/arXiv.2406.06700. URL https://proceedings.mlr.press/v235/vani24a.html.

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 (ICML)*, 2026. URL https://openreview.net/forum?id=milZYiwxSI.

Kaiyue Wen, Zhiyuan Li, and Tengyu Ma. Sharpness minimization algorithms do not only minimize sharpness to achieve better generalization, 2023. URL https://arxiv.org/abs/2307.11007.

Ruipeng Zhang, Ya-Chien Chang, and Sicun Gao. When maximum entropy misleads policy optimization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74617–74638. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhang25i.html.

Yang Zhao, Hao Zhang, and Xiuyuan Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 26982–26992. PMLR, 2022. URL https://proceedings.mlr.press/v162/zhao22i.html.

Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha Dvornek, Sekhar Tatikonda, James Duncan, and Ting Liu. Surrogate gap minimization improves sharpness-aware training. In *International Conference on Learning Representations (ICLR)*, 2022. doi: 10.48550/arXiv.2203.08065. URL https://arxiv.org/abs/2203.08065.

AI USE STATEMENT

The authors developed the research question and made the final methodological and scientific decisions. Generative AI tools assisted with literature discovery and organization; brainstorming hypotheses; drafting and editing portions of the manuscript, including translation and proofreading; discussing proof steps and checking mathematical derivations; and developing and debugging analysis, evaluation, and plotting code. The authors checked the mathematical arguments, verified citations against the original publications, and reviewed AI-assisted code against experiment logs and execution artifacts. AI outputs were treated as suggestions rather than scientific evidence. The authors take full responsibility for the paper's claims, results, and supplementary materials.

REPRODUCIBILITY STATEMENT

Mathematical derivations are provided in Appendix A, and complete training configurations, evaluation protocols, statistical procedures, and intervention definitions are given in Appendix B. Appendices C–F report the complete results and scope extensions behind the main-text tables and figures. Core experiments use 15 independent training seeds, with paired statistics over shared seeds where applicable. Anonymized code, configuration files, and analysis scripts for reproducing the reported experiments, evaluations, and analyses are provided in the supplementary material.

APPENDIX CONTENTS

A Mathematical Analysis **14**
A.1 Complete decomposition and proof of Proposition 1 14
A.2 Relative dose, absolute parameter dose, and action response 16
B Experimental and Measurement Protocols **17**
B.1 Training configuration, evaluation, and statistics 17
B.2 Training interventions and operational definitions 19
B.3 Parameter landscapes, common RMS, and displacement matching 21
C Training Scale and Action Robustness **23**
C.1 Full scale history replay 23
C.2 Exogenous fixed scale 25
C.3 Training scale and loss-sharpness readings 26
D Scale Feedback and Explicit Control **27**
D.1 Routing interventions 27
D.2 Explicit scale control and entropy calibration 28
E Endpoint Geometry Under Fixed Scale **29**
E.1 Mean updater, parameter dose, and behavior 29
E.2 Common action-displacement budget 29
E.3 L2-regularized endpoint geometry 30
F Scope and Sensitivity **32**
F.1 Box2D tasks 32
F.2 Performance under other perturbations 33
F.3 Checkpoint and measurement boundaries 37

A Mathematical Analysis

A.1 Complete decomposition and proof of Proposition 1

Objective function and diagonal Gaussian policy. We analyze policy updates on a fixed mini-batch of m transitions indexed by $t = 1, \ldots, m$, with states s_t, pre-clipping sampled actions a_t, and generalized advantage estimates (GAE) $\widehat{A}_t$. These quantities and the old policy are held fixed. We work in a neighborhood where the surrogate branches remain unchanged, parameter bounds are inactive, and the clean gradient norm is positive. The PPO actor loss is defined as the negative sample average of the clipped surrogate objective:

$$L(\theta) = -\frac{1}{m}\sum_{t=1}^{m} \min\Big(r_t(\theta)\widehat{A}_t,\ \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\widehat{A}_t\Big),$$

Let ℓ_t denote the clipped surrogate term, including the factor $1/m$, so that $L = -\sum_t \ell_t$. The probability ratio is $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\mathrm{old}}(a_t|s_t)}$. The policy is parameterized as an n-dimensional diagonal Gaussian distribution over continuous actions. The parameter vector $\theta = (\psi, \phi)$ consists of:

1. Mean-network parameters ψ (weights and biases), producing the deterministic action mean $\mu(s;\psi) \in \mathbb{R}^n$;
2. State-independent log-scale parameters $\phi \in \mathbb{R}^n$, with standard deviation $\sigma_j = e^{\phi_j}$ along action dimension $j \in \{1, \ldots, n\}$.

The conditional action density factorizes across dimensions:

$$\pi_\theta(a_t \mid s_t) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{(a_{t,j} - \mu_j(s_t))^2}{2\sigma_j^2}\right).$$

Defining the action residual relative to the current mean as $u_{t,j} = a_{t,j} - \mu_j(s_t)$, the log-likelihood is

$$\log \pi_\theta(a_t \mid s_t) = -\sum_{j=1}^{n}\left(\phi_j + \frac{1}{2}\log(2\pi) + \frac{u_{t,j}^2}{2e^{2\phi_j}}\right).$$

Log-likelihood gradients and clean actor gradients. For sample t, the analytical gradients of the log-likelihood with respect to ψ and ϕ_j are

$$\nabla_\psi \log \pi_\theta(a_t \mid s_t) = \sum_{j=1}^{n} \frac{a_{t,j} - \mu_j(s_t)}{\sigma_j^2} \nabla_\psi \mu_j(s_t) = \sum_{j=1}^{n} \sigma_j^{-2} u_{t,j} \nabla_\psi \mu_j(s_t),$$

$$\frac{\partial \log \pi_\theta(a_t \mid s_t)}{\partial \phi_j} = -1 - \frac{u_{t,j}^2}{2}(-2e^{-2\phi_j}) = \frac{u_{t,j}^2}{\sigma_j^2} - 1.$$

By the chain rule, define $q_t = \partial \ell_t / \partial \log r_t$ (including the $1/m$ factor): on active unclipped branches, $q_t = r_t \widehat{A}_t / m$; on clipped constant branches, $q_t = 0$. The clean gradients of the PPO loss with respect to ψ and ϕ are

$$g_\psi = \nabla_\psi L(\theta) = -\sum_{t=1}^{m} q_t \nabla_\psi \log \pi_\theta(a_t \mid s_t) = -\sum_{j=1}^{n} \sigma_j^{-2}\left(\sum_{t=1}^{m} q_t u_{t,j} \nabla_\psi \mu_j(s_t)\right),$$

$$g_{\phi,j} = \frac{\partial L(\theta)}{\partial \phi_j} = -\sum_{t=1}^{m} q_t \left(\frac{u_{t,j}^2}{\sigma_j^2} - 1\right).$$

Define the auxiliary vectors $z_j = \sum_{t=1}^{m} q_t u_{t,j} \nabla_\psi \mu_j(s_t)$ and $S = \sum_{j=1}^{n} \sigma_j^{-2} z_j$. The clean mean gradient is compactly written as $g_\psi = -S$, and the contribution from action dimension j is $g_{\psi,j} = -\sigma_j^{-2} z_j$, satisfying $g_\psi = \sum_{j=1}^{n} g_{\psi,j}$.

Scale dependence in the probability ratio. Let $G = \|\nabla_\theta L\|_2 = \sqrt{\|g_\psi\|_2^2 + \|g_\phi\|_2^2} = \sqrt{S^\top S + g_\phi^\top g_\phi} > 0$. When differentiating S with respect to log-scale ϕ_j, although states s_t, residuals $u_{t,j}$, and mean Jacobians $\nabla_\psi \mu$ do not explicitly depend on scale, the probability ratio r_t depends on ϕ_j. Consequently, q_t depends on ϕ_j:

$$\partial_{\phi_j} q_t = q_t \frac{\partial \log \pi_\theta(a_t \mid s_t)}{\partial \phi_j} = q_t \left(\frac{u_{t,j}^2}{\sigma_j^2} - 1 \right).$$

Thus, the derivative of z_i with respect to ϕ_j is

$$\partial_{\phi_j} z_i = \sum_{t=1}^{m} q_t \left(\frac{u_{t,j}^2}{\sigma_j^2} - 1 \right) u_{t,i} \nabla_\psi \mu_i(s_t).$$

Using $S = \sum_{i=1}^n \sigma_i^{-2} z_i$ and $\partial_{\phi_j}(\sigma_j^{-2}) = \partial_{\phi_j}(e^{-2\phi_j}) = -2\sigma_j^{-2}$, differentiating S gives

$$\partial_{\phi_j} S = -2\sigma_j^{-2} z_j + \sum_{i=1}^{n} \sigma_i^{-2} \partial_{\phi_j} z_i.$$

Three-term decomposition of SAM feedback. Under SAM perturbation radius ρ, the first-order local expansion of the update gradient is $\nabla L(\theta + \rho \frac{g}{G}) \approx g + \rho \nabla_\theta G$. The additional feedback term received by gradient descent is $-\rho \nabla_\theta G$. Projecting onto log-scale coordinate ϕ_j:

$$F_j = -\rho \partial_{\phi_j} G = -\rho \frac{\partial_{\phi_j}(G^2)}{2G} = -\frac{\rho}{G} \left(S^\top \partial_{\phi_j} S + g_\phi^\top \partial_{\phi_j} g_\phi \right).$$

Substituting the expansion of $\partial_{\phi_j} S$, F_j decomposes exactly into three components:

$$F_j = F_{\text{cross},j} + F_{\text{ratio},j} + F_{\phi,j},$$

where

$$F_{\text{cross},j} = \frac{2\rho}{G\sigma_j^2} z_j^\top S = \frac{2\rho}{G} \langle g_\psi, g_{\psi,j} \rangle,$$

$$F_{\text{ratio},j} = -\frac{\rho}{G} S^\top \sum_{i=1}^{n} \sigma_i^{-2} \partial_{\phi_j} z_i, \qquad F_{\phi,j} = -\frac{\rho}{G} g_\phi^\top \partial_{\phi_j} g_\phi.$$

Non-negative coordinate sum and directional entropy derivative. Summing $F_{\text{cross},j}$ across all action coordinates $j = 1, \ldots, n$:

$$\sum_{j=1}^{n} F_{\text{cross},j} = \frac{2\rho}{G} \left(\sum_{j=1}^{n} \sigma_j^{-2} z_j \right)^\top S = \frac{2\rho}{G} S^\top S = \frac{2\rho}{G} \|g_\psi\|_2^2 \geq 0.$$

Thus, the cross feedback transmitted from the mean perturbation to the log-scale parameters has a coordinate sum that is non-negative whenever the mean gradient is non-zero, scaled proportionally to $\|g_\psi\|_2^2$.

This non-negative sum has a direct information-theoretic interpretation. The differential entropy of an n-dimensional diagonal Gaussian policy is

$$H(\pi_\theta(\cdot \mid s_t)) = \frac{n}{2}(1 + \log(2\pi)) + \sum_{j=1}^{n} \phi_j.$$

Since $\partial H / \partial \phi_j = 1$ for all j, the gradient of entropy with respect to log-scale is the all-ones vector, $\nabla_\phi H = \mathbf{1} = (1, \ldots, 1)^\top$. Thus, the aggregate non-negativity of the cross term corresponds geometrically to a non-negative directional derivative of policy differential entropy along the cross-feedback direction:

$$\nabla_\phi H^\top F_{\text{cross}} = \sum_{j=1}^{n} F_{\text{cross},j} = \frac{2\rho}{G} \|g_\psi\|_2^2 \geq 0.$$

The cross component therefore has a non-negative entropy directional derivative. Individual coordinates can have either sign, and the remaining feedback components can change the total entropy direction.

Scope of the local expansion. The derivation assumes local smoothness of the loss surface at the current point. If the Hessian is locally Lipschitz continuous, the remainder of the SAM gradient expansion is $O(\rho^2)$; under standard twice continuous differentiability, the remainder is $o(\rho)$. The expansion holds within fixed smooth surrogate branches. Gradient clipping, Adam preconditioning, and parameter bounds determine how the local gradient translates into an actual update; the coordinate-sum identity alone does not imply entropy growth during training. The finite-radius routing experiments in Appendix D.1 test the full feedback channel rather than injecting only the analytical cross component.

A.2 Relative dose, absolute parameter dose, and action response

Filter-wise normalization and perturbation directions. Our 2D landscape evaluations use filter-wise normalization to express perturbations relative to each weight row (Li et al., 2018). Let the mean network parameters be $\psi = \{W_1, b_1, W_2, b_2, W_3, b_3\}$, where $W_{\ell,j}$ denotes the j-th row of weight matrix W_ℓ (the fan-in weight vector of neuron j in layer ℓ), with bias b_ℓ.

Two perturbation directions d_1, d_2 are generated by sampling independent Gaussian vectors $z_1, z_2 \sim \mathcal{N}(0, I)$. After global orthogonalization, filter-wise normalization is applied to each weight row:

$$d_{k,\ell,j} = \frac{z_{k,\ell,j}}{\|z_{k,\ell,j}\|_2 + \varepsilon} \|W_{\ell,j}\|_2, \qquad k \in \{1, 2\},$$

where $\varepsilon = 10^{-12}$ prevents division by zero, and bias perturbations are zeroed ($\Delta b_\ell = 0$). Here z_1, z_2 denote the directions after global orthogonalization. Although these directions are orthogonal, non-uniform row scaling means that the resulting global directions d_1, d_2 are generally non-orthogonal in parameter space.

Closed-form full-grid RMS parameter dose. On the grid $\mathcal{Q}$ spanned by d_1 and d_2, each point $q = (\alpha_{1,q}, \alpha_{2,q})$ specifies a parameter perturbation $\delta\psi_q = \alpha_{1,q} d_1 + \alpha_{2,q} d_2$. Omitting ε, the squared norm of each normalized direction equals the sum of squared Frobenius norms across all weight matrices:

$$\|d_k\|_2^2 = \sum_{\ell=1}^{3} \sum_{j} \|W_{\ell,j}\|_2^2 = \sum_{\ell=1}^{3} \|W_\ell\|_F^2, \qquad k \in \{1, 2\}.$$

The root-mean-square (RMS) absolute parameter perturbation across the grid $\mathcal{Q}$ is defined as:

$$D_{\text{rms}} = \left(\frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \|\delta\psi_q\|_2^2 \right)^{1/2}.$$

Expanding the squared Euclidean norm:

$$\|\delta\psi_q\|_2^2 = \alpha_{1,q}^2 \|d_1\|_2^2 + \alpha_{2,q}^2 \|d_2\|_2^2 + 2\alpha_{1,q}\alpha_{2,q}\langle d_1, d_2\rangle.$$

Our evaluation uses a symmetric 11×11 grid over $[-0.20, 0.20]^2$ with step size 0.04 ($|\mathcal{Q}| = 121$). By grid symmetry:

1. The cross-moment vanishes identically: $\frac{1}{|\mathcal{Q}|} \sum_q \alpha_{1,q}\alpha_{2,q} = 0$;
2. The second moments along each axis are $\frac{1}{|\mathcal{Q}|} \sum_q \alpha_{1,q}^2 = \frac{1}{|\mathcal{Q}|} \sum_q \alpha_{2,q}^2 = \frac{1}{11} \sum_{i=-5}^{5} (0.04 \times i)^2 = 0.016$.

Because the cross-term vanishes in the grid average regardless of whether d_1 and d_2 are orthogonal, substituting the second moments gives the following identity when the numerical stabilizer ε is omitted:

$$D_{\text{rms}} = \sqrt{0.016\|d_1\|_2^2 + 0.016\|d_2\|_2^2} = \sqrt{0.032} \left(\sum_{\ell=1}^{3} \|W_\ell\|_F^2 \right)^{1/2}. \tag{3}$$

This identity shows that under filter normalization, identical relative grid coordinates ($\alpha \in [-0.2, 0.2]$) impose an absolute parameter perturbation that scales directly with the Frobenius norm of the model's weights. When SAM induces smaller weight norms, filter normalization automatically injects smaller absolute parameter perturbations.

Propagation to action response in 3-layer Tanh networks. Parameter perturbations affect policy performance through their induced action displacement. Consider the 3-layer mean network taking normalized state $x = h_0$:

$$h_1 = \tanh(W_1 x + b_1), \quad h_2 = \tanh(W_2 h_1 + b_2),$$
$$h_3 = \tanh(W_3 h_2 + b_3), \quad \mu(x) = a_{\max} h_3.$$

where $a_{\max}$ scales the bounded action space. Let $D_\ell = \mathrm{diag}(1 - h_\ell^2)$ be the diagonal Jacobian of the tanh activation at layer ℓ. Since the derivative of tanh is bounded by 1, $\|D_\ell\|_2 \le 1$.

For weight perturbations ΔW_ℓ with $\Delta b_\ell = 0$, define layer input perturbations $e_\ell = \Delta W_\ell h_{\ell-1}$. A first-order Taylor expansion yields:

$$\Delta\mu(x) = a_{\max} D_3 \left[e_3 + W_3 D_2 e_2 + W_3 D_2 W_2 D_1 e_1\right] + O(\|\Delta W\|_2^2).$$

Applying the sub-multiplicative property and $\|D_\ell\|_2 \le 1$, the first-order action displacement is bounded by:

$$\|\Delta\mu(x)\|_2 \le a_{\max} \Big[\|\Delta W_3\|_2 \|h_2\|_2 + \|W_3\|_2 \|\Delta W_2\|_2 \|h_1\|_2 + \|W_3\|_2 \|W_2\|_2 \|\Delta W_1\|_2 \|x\|_2 \Big] + O(\|\Delta W\|_2^2).$$

Using $\|\Delta W_\ell\|_2 \le \|\Delta W_\ell\|_F$, this bound clarifies how perturbations propagate:

1. Input-layer perturbations (ΔW_1) are cascaded and amplified by downstream weight norms $\|W_2\|_2 \|W_3\|_2$;
2. Output-layer perturbations (ΔW_3) directly affect the action without downstream amplification;
3. The quantity $a_{\max} \prod_{\ell=1}^{3} \|W_\ell\|_2$ bounds sensitivity to the normalized input x, but does not equal the parameter Jacobian norm, as perturbations are injected at different depths.

Disentangling dose from robustness. The local response depends on the applied parameter perturbations, hidden activations, and downstream weights. Filter-wise normalization also makes the absolute perturbation dose depend on the endpoint weights, as shown in Equation 3. These relations motivate measuring dose and action displacement jointly; a smaller upper bound alone does not determine which of two trained policies has the smaller actual displacement. Appendix B.3 specifies the common-RMS evaluation and subsequent displacement matching used to examine their relationship with return retention.

B Experimental and Measurement Protocols

B.1 Training configuration, evaluation, and statistics

Core experiments are conducted across four standard MuJoCo-v4 continuous control benchmarks, using 15 independent random training seeds for each condition over 3 million (3M) environment steps. The main text strictly evaluates the final checkpoint at the end of training; sensitivity analyses on best checkpoints are reported in Appendix F.3 and Table 23. The core policy optimization objective follows the clipped surrogate loss L_{PPO} defined in Appendix A.1. Native PPO and SAM-PPO set the entropy regularization coefficient to zero, while explicit entropy and weight-norm regularization append their respective penalty terms directly to this baseline objective. Table 5 lists the complete hyperparameters and implementation specifications.

Table 5: Base training configuration and hyperparameters.

Hyperparameter / Setting	Specification / Value
Environment & Evaluation	
Core environments	HalfCheetah-v4, Hopper-v4, Walker2d-v4, Ant-v4
Training budget	15 independent random seeds, 3M environment steps
Terminal evaluation	50 deterministic evaluation episodes at final 3M-step checkpoint
Online validation	10 episodes every 10k steps (training monitoring only)
Policy Architecture	
Network architecture	2-layer MLP (64 units per hidden layer, $\tanh$ activations)
Mean action output	Linear layer with $\tanh$ squashing to $[-a_{\max}, a_{\max}]$
Policy parameterization	Diagonal Gaussian $\mathcal{N}(\mu(s), \mathrm{diag}(\sigma^2))$, state-independent $\log \sigma$
Scale initialization & bounds	Initial $\phi_j = -0.5$ ($\sigma_j \approx 0.61$), clamped to $[\log(10^{-3}), \log(10)]$
Weight initialization & bounds	Orthogonal init (actor output gain 0.01, others 1.0), weights in $[-5, 5]$
Optimization & PPO	
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-5}$)
Learning rate	$\eta = 3 \times 10^{-4}$ for actor and critic, linearly annealed to 0
PPO discount & GAE	Discount factor $\gamma = 0.99$, GAE parameter $\lambda_{\mathrm{GAE}} = 0.95$
PPO clipping	Surrogate objective clipping parameter $\epsilon_{\mathrm{clip}} = 0.2$
Rollout & update	Buffer horizon $T = 2048$, minibatch size 64, 10 epochs per update
Gradient clipping	Global L_2 norm clipped at 0.5 (actor and critic separately)
Normalizations	Online running state normalization, return reward scaling
Algorithmic Interventions	
Native SAM perturbation	Actor-only, radius $\rho = 0.008$, joint perturbation on (ψ, ϕ)
Fixed-scale controls	Constant $\sigma \in \{0.05, 0.10, 0.20, 0.40, 0.80\}$, ϕ gradient disabled

The final tanh squashing on mean actions and the environment action clipping are two distinct operations. In rollouts, PPO logs the *unclipped sampled Gaussian action* to compute the probability ratio $r_t(\theta)$ under standard Gaussian likelihoods, while the clipped action $\mathrm{clip}(\widetilde{a}_t, -a_{\max}, a_{\max})$ is executed in the environment simulator. Post-training evaluations disable exploration noise and evaluate the deterministic mean action $\mu(s; \psi)$ across 50 full episodes.

External action noise is injected at test time as $a_t = \mathrm{clip}(\mu(s_t) + \epsilon_t, -a_{\max}, a_{\max})$ with $\epsilon_t \sim \mathcal{N}(0, \tau^2 I)$. Noise magnitude τ is evaluated at 11 uniformly spaced levels in $[0, 0.5]$ over 50 episodes per level. For each training seed, the area under the noise-response curve (AUC) is computed via trapezoidal integration, and normalized retention is given by:

$$\mathrm{nAUC} = \frac{\mathrm{AUC}}{0.5 J_{\mathrm{noise}}(0)},$$

where $J_{\mathrm{noise}}(0)$ is the zero-noise return from the same noise-evaluation stream. nAUC is computed per training seed and then averaged. For a non-positive denominator, we report raw returns and AUC rather than interpreting the ratio as retention. We distinguish this baseline from the independently evaluated nominal return J_{nom} and the parameter-grid baseline $J_{\mathrm{geom}}(0)$; these three measurements are not substituted for one another.

Unless stated otherwise, tables report means $\pm$ SE across 15 training seeds. Paired effects are computed within seed before aggregation; their SE is the sample standard deviation of the 15 differences divided by $\sqrt{15}$. Figure captions specify whether bands show SE or 1.96 SE. Where bootstrap intervals are reported, they are percentile intervals obtained by resampling training seeds. Evaluation episodes and parameter-grid points are not independent training samples. Pooled summaries across tasks are descriptive rather than estimates over a population of tasks.

Experiments were conducted across Linux and Windows batches. Supplementary final-checkpoint action-noise evaluations used Python 3.12.14, PyTorch 2.13.0+cpu, Gymnasium 1.3.0, and MuJoCo 3.10.0; numerical library thread environment variables were set to 1, with PyTorch intra-op threads set to 1 and interop threads to 8. Checkpoints and evaluation records retain all available environment metadata. Action and state perturbations from Robust-Gymnasium constitute an independent extension evaluated in Appendix F.2 and are not conflated with the primary action-noise protocol.

B.2 Training interventions and operational definitions

Exogenous scale history replay protocol. During donor training, the elementwise log-scale parameters $\{\phi_i^{\text{donor}}\}_{i=0}^{K}$ are recorded at each rollout update boundary. In replay experiments, the receiver mean and value networks are initialized from scratch and interact independently with the environment to collect their own trajectories, without reusing donor states, actions, or rollout buffers; the actor scale is initialized from ϕ_0^{donor} and then follows the donor schedule. Scale parameters are written strictly at update cycle boundaries as a piecewise-constant schedule, without intra-rollout interpolation. Throughout all epochs and minibatches within each update cycle, scale parameters are detached from the actor optimizer and frozen completely ($\text{requires_grad} \leftarrow \text{False}$). Within the actor, only mean-network parameters ψ (weights and biases) are optimized, while the critic is trained normally. The unified procedure is formalized in Algorithm 1.

Algorithm 1 Exogenous Scale-History Replay

Require: Donor scale history $\{\phi_i^{\text{donor}}\}_{i=0}^{K}$, environment $\mathcal{E}$, updater rule $\mathcal{U} \in \{\text{PPO}, \text{Mean-SAM}\}$, rollout horizon T, epochs K_{epochs}, minibatch size M

Ensure: Trained receiver policy $\pi_{(\psi, \phi_K^{\text{donor}})}$

1: **Validate donor trajectory:** Verify environment, training seed, donor identity, action-vector dimension, monotone update indices, and full update coverage; stop on inconsistency
2: **Initialize receiver:** Fresh parameters $\psi \leftarrow \psi_0$, initial scale $\phi \leftarrow \phi_0^{\text{donor}}$
3: **Isolate scale parameter:** Remove ϕ from actor optimizer and its optimizer state; clear its gradient and disable gradient tracking
4: **for** rollout iteration $i = 0, 1, \ldots, K-1$ **do**
5: Write boundary scale: $\phi \leftarrow \phi_i^{\text{donor}}$ {Piecewise-constant schedule}
6: Collect rollout transitions $\mathcal{D}_i = \{(s_t, a_t, r_t, s_{t+1})\}_{t=1}^{T}$ in $\mathcal{E}$ with current receiver $\pi_{(\psi,\phi)}$
7: Compute GAE advantages $\widehat{A}_t$ and returns on $\mathcal{D}_i$
8: Freeze scale ϕ across all K_{epochs} epochs and minibatches of buffer $\mathcal{D}_i$
9: Update mean parameters $\psi \leftarrow \mathcal{U}(\psi; \mathcal{D}_i)$ {PPO or Mean-SAM}
10: Update critic parameters θ_v by baseline MSE loss on $\mathcal{D}_i$
11: Advance boundary: write $\phi \leftarrow \phi_{i+1}^{\text{donor}}$
12: **end for**
13: **return** Final receiver checkpoint $(\psi, \phi_K^{\text{donor}})$

Routing with matching source dose and clean-gradient clipping. To implement isolated block-wise parameter perturbations, two confounding coupling mechanisms must be disentangled:

1. **Perturbation dose coupling:** In native SAM, the perturbation radius ρ is allocated across the joint parameter space (ψ, ϕ), so the dose received by each block depends on the relative gradient magnitude of the other block.
2. **Global clipping cross-talk:** PPO applies global gradient norm clipping. If perturbed gradients alter the total gradient norm, the global clipping factor changes, inadvertently modifying the effective step size of unperturbed parameter blocks.

To decouple these mechanisms, we impose two controls:

- **Fixed source perturbation dose:** For a chosen source block $a \in \{\psi, \phi\}$, its perturbation magnitude is fixed strictly to the nominal radius ρ:

$$\delta_a = \rho \frac{g_a}{\|g_a\|_2},$$

  ensuring identical perturbation intensity regardless of routing configuration.
- **Clean-gradient clipping invariance and norm matching:** The global clipping factor c is computed strictly from the unperturbed clean gradient $g = (g_\psi, g_\phi)$:

$$c = \min\left(1, \frac{0.5}{\|g\|_2 + 10^{-6}}\right).$$

For a receiving block $b \in \{\psi, \phi\}$, candidate perturbed gradient $h_b = \nabla_b L(\theta + \delta_a)$ is rescaled so its norm strictly matches the unperturbed clipped norm:

$$\widetilde{g}_b = c\|g_b\|_2 \frac{h_b}{\|h_b\|_2}, \qquad \widetilde{g}_{\bar{b}} = c g_{\bar{b}}.$$

Under this protocol, the gradient norm of each block passed to Adam strictly matches its clean unperturbed counterpart, and the shared clipping factor is not recomputed from perturbed gradients. Across different routing trajectories, clean gradients and optimizer states continue to diverge naturally. Algorithm 2 formalizes the complete four-phase execution per minibatch.

Algorithm 2 Matched-Dose Feedback Routing with Clean-Norm Control

Require: Minibatch data $\mathcal{B}$, current parameters $\theta = (\psi, \phi)$, perturbation radius ρ, source block $a \in \{\psi, \phi\}$, receiver block $b \in \{\psi, \phi\}$, clipping threshold $G_{\max} = 0.5$

Ensure: Updated parameters θ after one routed step under clean-norm control

Phase 1: Clean gradient pass
  Compute clean actor gradient $g = (g_\psi, g_\phi) \leftarrow \nabla_\theta L_{\mathrm{PPO}}(\theta; \mathcal{B})$
  Lock clean clipping factor: $c \leftarrow \min\left(1, \frac{G_{\max}}{\|g\|_2 + 10^{-6}}\right)$
  Snapshot clean parameters: $\theta_{\mathrm{snap}} \leftarrow \theta$
Phase 2: Source perturbation (matched dose)
  Require a finite, non-degenerate source norm $\|g_a\|_2 > 0$; otherwise stop the routed update
  Apply source perturbation: $\delta_a \leftarrow \rho \frac{g_a}{\|g_a\|_2}$ and set $\theta_a \leftarrow \theta_a + \delta_a$
Phase 3: Perturbed pass and restoration
  Compute perturbed gradient on the *same* minibatch: $h_b \leftarrow \nabla_b L_{\mathrm{PPO}}(\theta; \mathcal{B})$
  Restore clean parameters $\theta \leftarrow \theta_{\mathrm{snap}}$ before constructing the routed update
Phase 4: Clean-norm assembly and step
  If $\|h_b\|_2 \le 10^{-12}$, keep a zero receiver gradient when $c\|g_b\|_2 \le 10^{-12}$; otherwise stop
  Otherwise form $\widetilde{g}_b \leftarrow c\|g_b\|_2 \frac{h_b}{\|h_b\|_2}$ {Preserves clean block norm}
  Form unrouted block gradient: $\widetilde{g}_{\bar{b}} \leftarrow c g_{\bar{b}}$ {Unperturbed clean gradient}
  Apply actor update: $\theta \leftarrow \mathrm{Adam}(\theta, \widetilde{g})$ {No recomputed shared clipping}
  Update critic network separately; apply the implementation bounds to mean-network weights and log-scale parameters
return Updated parameters θ

RFD loss and one-update-lag schedule. Regularized feedback descent (RFD) adds an inverse-variance scale penalty to the PPO actor loss, explicitly constructing scale-restoring feedback that strengthens as scale contracts:

$$L_{\mathrm{RFD}}(\theta) = L_{\mathrm{PPO}}(\theta) + \frac{\lambda_k}{2} \sum_{j=1}^{n} \sigma_j^{-2},$$

where $\sigma_j = e^{\phi_j}$ is the standard deviation along the j-th action dimension. The scalar coefficient λ_k follows a *one-update-lag schedule*: for rollout update cycle k, λ_k takes the average coupling intensity measured during the previous cycle $k-1$ as a constant input ($\lambda_1 = 0$ for the initial cycle):

$$c_{k,b} = \frac{\sigma_{\mathrm{GM},k,b}^2 \|g_{\psi,k,b}\|_2^2}{\sqrt{n}\, G_{\mathrm{clean},k,b}}, \qquad C_k = \frac{1}{M_k} \sum_{b=1}^{M_k} c_{k,b}, \qquad \lambda_1 = 0, \quad \lambda_k = \frac{2\rho}{\sqrt{n}} C_{k-1} \quad (k \ge 2),$$

where $\sigma_{\mathrm{GM},k,b} = \exp\left(\frac{1}{n}\sum_j \phi_{j,k,b}\right)$ is the geometric mean scale on minibatch b of update k, $g_{\psi,k,b}$ is the corresponding clean mean gradient, and $G_{\mathrm{clean},k,b} = \sqrt{\|g_{\psi,k,b}\|_2^2 + \|g_{\phi,\mathrm{clean},k,b}\|_2^2 + 10^{-12}}$ is the total clean gradient norm.

Single backward pass and clean statistic recovery. Unlike native SAM which requires two backward passes per minibatch, RFD executes only a *single backward pass*. Because the penalty's gradient with respect to log-scale has the simple analytic form $\nabla_\phi[\frac{\lambda_k}{2}\sum_j e^{-2\phi_j}] = -\lambda_k \sigma^{-2}$, subtracting

this known analytic contribution prior to gradient clipping ($g_{\phi,\text{clean}} = \nabla_\phi L_{\text{RFD}} + \lambda_k \sigma^{-2}$) exactly recovers the clean scale gradient at zero extra computational cost. Coupling statistics $c_{k,b}$ are detached from the autograd graph and stored as pure numerical constants for cycle $k+1$. Algorithm 3 formalizes the complete execution procedure.

Explicit entropy baseline and calibration protocol. To test whether generic explicit scale maintenance achieves comparable robustness gains, policy entropy regularization is evaluated as an explicit baseline: $L_{\text{PPO}} - \beta\mathcal{H}(\pi)$. To avoid selection bias toward favorable outcomes, the entropy coefficient β is calibrated strictly by matching the scale trajectory of SAM-PPO rather than by optimizing action-noise AUC:

1. Candidates are evaluated over a discrete grid $\beta \in \{0.00125, 0.0025, 0.005, 0.0075, 0.01, 0.0125\}$.
2. Across 10 calibration seeds, the root-mean-square error (RMSE) between the geometric mean scale trajectory $\sigma_{\text{GM}}(t)$ and that of same-seed SAM-PPO is computed at the recorded SAM environment steps after linear interpolation. Seed-level RMSEs are averaged, resolving ties by the smaller coefficient.
3. The coefficient minimizing mean RMSE is selected as the calibrated value β^*: $\beta^* = 0.0075$ for HalfCheetah, Hopper, and Ant, and $\beta^* = 0.005$ for Walker2d. The selected coefficients and their trajectory RMSEs, together with behavioral noise-response curves, are reported in Appendix D.2 (Table 11 and Figure 5).
4. With β^* fixed, final performance and action-noise evaluations are reported across the full 15 seeds, including the 10 calibration seeds.

Algorithm 3 One-Update-Lag RFD

Require: Policy $\pi_{(\psi,\phi)}$, rollout buffer $\mathcal{D}_k$, previous lag statistic C_{k-1} ($C_0 = 0$), action dimension n, radius ρ, clipping threshold $G_{\max} = 0.5$

Ensure: Updated policy parameters and next-update lag statistic C_k

Phase 1: Initialize update coefficient
  Fix scalar weight: $\lambda_k \leftarrow \frac{2\rho}{\sqrt{n}} C_{k-1}$ {$\lambda_1 = 0$; detached scalar across rollout k}
  Initialize minibatch measurement buffer: $\mathcal{M} \leftarrow \emptyset$
Phase 2: Minibatch updates and clean gradient recovery
for each epoch and minibatch $\mathcal{B} \subset \mathcal{D}_k$ **do**
  Evaluate combined surrogate loss: $L_{\text{total}} \leftarrow L_{\text{PPO}}(\psi, \phi; \mathcal{B}) + \frac{\lambda_k}{2} \sum_{j=1}^{n} e^{-2\phi_j}$
  Execute single backward pass on L_{total}
  Recover clean scale gradient: $g_{\phi,\text{clean}} \leftarrow \nabla_\phi L_{\text{total}} + \lambda_k e^{-2\phi}$ {Pre-clip recovery}
  Geometric scale: $\sigma_{\text{GM}} \leftarrow \exp\left(\frac{1}{n}\sum_{j=1}^{n} \phi_j\right)$
  Clean gradient norm: $G_{\text{clean}} \leftarrow \sqrt{\|g_\psi\|_2^2 + \|g_{\phi,\text{clean}}\|_2^2 + 10^{-12}}$
  Record coupling: $c_{k,b} \leftarrow \frac{\sigma_{\text{GM}}^2 \|g_\psi\|_2^2}{\sqrt{n}\, G_{\text{clean}}}$ and append to $\mathcal{M}$ {No autograd graph}
  Clip gradients to $\|g\|_2 \le G_{\max}$ and execute Adam step on actor
  Update critic network separately; apply the implementation bounds to mean-network weights and log-scale parameters
end for
Phase 3: Accumulate lag for next update
  Fold minibatch statistics: $C_k \leftarrow \frac{1}{|\mathcal{M}|} \sum_{c \in \mathcal{M}} c$ {Stored for update $k+1$}
return Updated policy $\pi_{(\psi,\phi)}$ and coefficient statistic C_k

B.3 Parameter landscapes, common RMS, and displacement matching

Evaluating policy flatness (return retention) in parameter space under fixed-radius perturbations faces two central confounding factors:

1. **Parameter-level confound (weight-norm differences):** Under filter-wise normalization, the same relative grid radius can induce different absolute RMS parameter doses when endpoint weight norms differ.
2. **Action-level confound (functional sensitivity differences):** Even with identical parameter perturbation doses, policies can differ substantially in how sensitively parameter shifts translate into action changes, potentially creating an illusion of flatness if action outputs barely move.

To address both confounds, we implement a two-tier controlled evaluation protocol.

Evaluation grids and baseline definition. For each checkpoint, following Appendix A.2, two i.i.d. standard-Gaussian directions are sampled, globally orthogonalized, and normalized row-by-row by weight norms, with bias perturbations set to zero; the normalized directions are generally no longer strictly orthogonal. Parameter perturbations are evaluated across an 11×11 grid (121 points) over $[-0.20, 0.20]^2$ with step size 0.04, with 50 evaluation episodes per grid point. For arm a, let $J_{\mathrm{geom},a}(0)$ denote its separately evaluated unperturbed grid baseline and define $R_a(q) = J_a(q)/J_{\mathrm{geom},a}(0)$. When this baseline is positive, full-grid retention is:

$$A_{80,a} = \frac{1}{121} \sum_{q \in \mathcal{Q}} \mathbf{1}\{R_a(q) \geq 0.8\}.$$

For a non-positive baseline, we report raw returns and do not interpret A_{80} as a retention metric. Simultaneously, we evaluate mean action displacement on 500 shared reference states $\mathcal{S}$ collected by the same-seed native PPO policy:

$$d(q) = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \|\mu_{\psi+\delta\psi(q)}(s) - \mu_\psi(s)\|_2.$$

Control 1: Common full-grid RMS parameter dose. To control for weight-norm disparities, we take the same-seed fixed-scale ($\sigma = 0.20$) PPO policy as reference P, with weight norm $N_a = (\sum_\ell \|W_{a,\ell}\|_F^2)^{1/2}$. For any comparison policy M, the grid radius is adaptively scaled by the ratio of their weight norms:

$$s_M = \frac{N_P}{N_M}, \qquad r_M = 0.20 \cdot s_M, \qquad r_P = 0.20. \tag{4}$$

By the derivation in Appendix A.2, the full-grid parameter RMS perturbation dose satisfies $D_{\mathrm{rms},a} = \sqrt{0.8}\, r_a N_a$. This radius scaling equalizes the overall RMS parameter dose across paired grids, up to numerical stabilization and radius rounding.

Control 2: Strict one-to-one action-displacement matching. To control for differences in realized action changes under parameter perturbations, we perform displacement matching on the common-RMS grids:

1. **Exclude center coordinate:** Center coordinates are removed from both grids, leaving 120 candidate perturbation points per side.
2. **Strict tolerance pairing:** With I_z denoting the interquartile range of the pooled 240 displacement values, candidate pairs are feasible if $|d_M(i) - d_P(j)| \leq 0.05 I_z$.
3. **Optimal bipartite matching:** Blinding to returns, linear assignment identifies a one-to-one matching $\mathcal{M}_z$ without replacement that maximizes matched pairs and minimizes total displacement difference.
4. **Matched-subset retention gap:** On the matched action-displacement subset, the paired return-retention difference is:

$$\Delta p_{80,z} = \frac{1}{|\mathcal{M}_z|} \sum_{(i,j) \in \mathcal{M}_z} \left[\mathbf{1}\{R_M(i) \geq 0.8\} - \mathbf{1}\{R_P(j) \geq 0.8\}\right]. \tag{5}$$

   This statistic compares retention under closely matched average action displacement amplitudes on shared reference states. It does not constrain displacement directions, state-wise patterns, or subsequent trajectory distributions; matched subsets no longer retain identical parameter RMS.

Sensitivity analyses evaluate tighter ($\eta = 0.025$) and wider ($\eta = 0.10$) tolerances, as well as equal-frequency binning. Algorithm 4 outlines the complete evaluation and matching procedure.

Algorithm 4 Common-RMS Evaluation and Displacement Matching

Require: Paired final checkpoints M and fixed-scale PPO P; paired direction and episode seeds; 500 shared raw reference states; tolerance fraction $\eta = 0.05$

Ensure: Full-grid A_{80} and D_{rms}; matched-subset difference Δp_{80}, matched indices, coverage, and displacement balance

for each task and paired training seed z **do**
  Generate paired raw directions; globally orthogonalize, then normalize each checkpoint's weight rows as in Appendix A.2; zero bias perturbations
  Set $N_a = (\sum_\ell \|W_{a,\ell}\|_F^2)^{1/2}$, $r_P = 0.20$, and $r_M = 0.20 N_P / N_M$
  for arm $a \in \{M, P\}$ **do**
    Freeze its observation normalization; load or evaluate its saved clean baseline $J_{\mathrm{geom},a}(0)$
    Evaluate the 11×11 grid of radius r_a: hold each perturbation fixed for 50 episodes, record $J_a(q)$ and reference-state displacement $d_a(q)$, and restore clean parameters after each point
    Record $D_{\mathrm{rms},a} = (121^{-1} \sum_q \|\delta\psi_a(q)\|_2^2)^{1/2}$ and $\bar{d}_a = 121^{-1} \sum_q d_a(q)$
    If either arm has a non-positive baseline, retain raw measurements and omit this pair from retention comparisons
    Set $R_a(q) = J_a(q)/J_{\mathrm{geom},a}(0)$ and $A_{80,a} = 121^{-1} \sum_q \mathbf{1}\{R_a(q) \geq 0.8\}$
    Exclude the center coordinate, leaving 120 matching candidates
  end for
  Pool both displacement arrays; compute $I_z = Q_{0.75} - Q_{0.25}$ and require $I_z > 0$
  Using displacements only, find a one-to-one matching $\mathcal{M}_z$ with $|d_M(i) - d_P(j)| \leq \eta I_z$: maximize pair count, then minimize total displacement difference
  Require $|\mathcal{M}_z| > 0$; compute $\Delta p_{80,z}$ using Equation 5
  Save pair indices, count, mean displacement gap, Wasserstein distance, and $\Delta p_{80,z}$
end for
Average the 15 seed-level differences equally; report SE and mean/minimum matched counts
Repeat with $\eta \in \{0.025, 0.10\}$ and the 5/10/20-bin weighting rules defined in Appendix B.3

C Training Scale and Action Robustness

C.1 Full Scale History Replay

The scale-history replay in Section 3.2 shows that the training scale systematically changes the final policy's action-noise robustness. This section reports the complete results of the 2×2 replay design, examining both factors and their interaction through paired comparisons that hold either the mean updater or the scale history fixed. All receiver policies are trained from scratch; the detailed replay protocol is given in Appendix B.2 and Algorithm 1.

Replay performance. Table 6 reports nominal return, action AUC, and normalized AUC (nAUC). Arrows in the table indicate "scale history source $\rightarrow$ receiver mean updater": for example, SAM $\rightarrow$ PPO denotes training a PPO receiver with SAM's scale history, and PPO $\rightarrow$ Mean-SAM denotes training a Mean-SAM receiver with PPO's scale history.

Table 6: **Complete scale-history replay.** Arrows denote scale-history donor → receiver mean updater; parentheses report nAUC.

Environment	Scale history → Updater	Nominal return	Action AUC (nAUC)
HalfCheetah	PPO → PPO	6136.6 ± 364.1	1294.6 ± 84.8 (42.1%)
	SAM → PPO	6712.9 ± 318.4	2030.1 ± 82.3 (61.3%)
	PPO → Mean-SAM	5065.0 ± 425.9	1089.7 ± 101.1 (45.3%)
	SAM → Mean-SAM	6365.7 ± 274.6	1995.2 ± 86.3 (63.6%)
Hopper	PPO → PPO	2937.3 ± 182.3	525.7 ± 26.5 (36.0%)
	SAM → PPO	3152.8 ± 210.6	830.6 ± 31.8 (56.8%)
	PPO → Mean-SAM	2883.5 ± 207.0	539.0 ± 24.8 (40.0%)
	SAM → Mean-SAM	3290.1 ± 171.3	836.6 ± 40.5 (52.4%)
Walker2d	PPO → PPO	4501.1 ± 193.6	850.0 ± 66.4 (38.0%)
	SAM → PPO	4774.5 ± 186.0	1798.4 ± 81.0 (74.0%)
	PPO → Mean-SAM	4575.3 ± 170.2	791.1 ± 42.9 (35.3%)
	SAM → Mean-SAM	4298.7 ± 201.3	1666.7 ± 73.0 (79.5%)
Ant	PPO → PPO	4512.9 ± 171.5	783.8 ± 30.3 (34.4%)
	SAM → PPO	5388.7 ± 92.1	1203.1 ± 18.4 (44.9%)
	PPO → Mean-SAM	4759.0 ± 141.1	745.2 ± 17.6 (31.4%)
	SAM → Mean-SAM	5616.7 ± 58.0	1238.3 ± 11.7 (43.9%)

Factorial paired contrasts: scale history, mean updater, and interaction. Table 7 evaluates the individual and joint effects of scale history and mean updater via matched paired differences within each seed:

Table 7: **Paired action-AUC contrasts in matched 2×2 scale replay.** Contrasts evaluate replacing PPO scale with SAM scale (top block) and replacing PPO updater with Mean-SAM (middle block). Pointwise 95% bootstrap CIs over 15 matched seeds.

Paired contrast	HalfCheetah	Hopper	Walker2d	Ant
SAM scale-history replacement				
Under PPO updater	$+735.5$ [490.0, 975.7]	$+304.9$ [221.6, 396.2]	$+948.4$ [768.0, 1111.2]	$+419.2$ [353.4, 485.3]
Under Mean-SAM updater	$+905.5$ [645.6, 1168.3]	$+297.6$ [200.3, 391.4]	$+875.6$ [727.6, 1027.8]	$+493.2$ [444.9, 537.4]
SAM mean-updater replacement				
Under PPO scale	-204.9 [−435.5, 41.5]	$+13.4$ [−65.2, 104.6]	-58.9 [−175.2, 48.7]	-38.7 [−100.3, 28.8]
Under SAM scale	-34.9 [−201.3, 124.2]	$+6.0$ [−114.2, 121.8]	-131.7 [−292.2, 5.1]	$+35.3$ [11.2, 59.1]
Interaction (Synergy)	$+170.0$ [−131.4, 463.8]	-7.4 [−179.7, 147.4]	-72.8 [−211.6, 56.9]	$+73.9$ [0.2, 144.4]

- **SAM scale-history replacement:** Holding the mean updater fixed, replacing the baseline scale history with SAM's policy scale consistently improves action AUC across all four benchmarks (all eight paired differences strictly exceed zero, ranging from $+297.6$ to $+948.4$). This improvement occurs whether the policy uses standard PPO or Mean-SAM. Furthermore, as demonstrated in Walker2d under Mean-SAM (nominal return shifts by -276.5 ± 273.8 while action AUC increases by $+875.6 \pm 79.3$), the robustness gains conferred by scale history do not require concurrent increases in nominal returns.
- **SAM mean-updater replacement:** Conversely, holding the scale history fixed and replacing standard PPO with SAM's mean updater (Mean-SAM) produces negligible or inconsistent

changes in action AUC (ranging from -204.9 to $+35.3$, with nearly all 95% bootstrap confidence intervals spanning zero).

- **Interaction (synergy):** The interaction effect—testing whether combining the SAM scale with the Mean-SAM updater creates super-additive synergy—is statistically indistinguishable from zero in three of the four benchmarks (spanning zero in HalfCheetah, Hopper, and Walker2d, and marginally positive in Ant at $+73.9$, 95% CI $[+0.2, +144.4]$).

Within the matched replay design, replacing the scale history produces larger and more consistent action-AUC gains than replacing the mean updater.

Self-replay ablation and native baseline comparison. The two self-replay conditions (PPO $\rightarrow$ PPO and SAM $\rightarrow$ Mean-SAM) serve as an ablation to inspect differences introduced by the replay protocol itself. Because receivers collect independent trajectories and Mean-SAM perturbs only mean parameters, self-replay exhibits numerical shifts relative to native training (for example, $+277.2$ in HalfCheetah and -193.7 in Walker2d for action AUC). Consequently, all main and interaction effects in Table 7 are evaluated strictly via paired comparisons within the four replay conditions, controlling for trajectory sampling differences. Performance of these replay policies under friction and mass variations is reported in Appendix F.2.

C.2 Exogenous Fixed Scale

The replay experiment in Appendix C.1 preserves both the temporal dynamics and the per-dimension variation of scale histories. To isolate the effect of the scale level itself, we fix the policy scale to a constant throughout training, optimize the mean policy with standard PPO, and evaluate action-noise responses across five preset scale levels. Table 8 summarizes nominal returns and action AUCs across all five levels, and Figure 4 displays the full action-noise response curves.

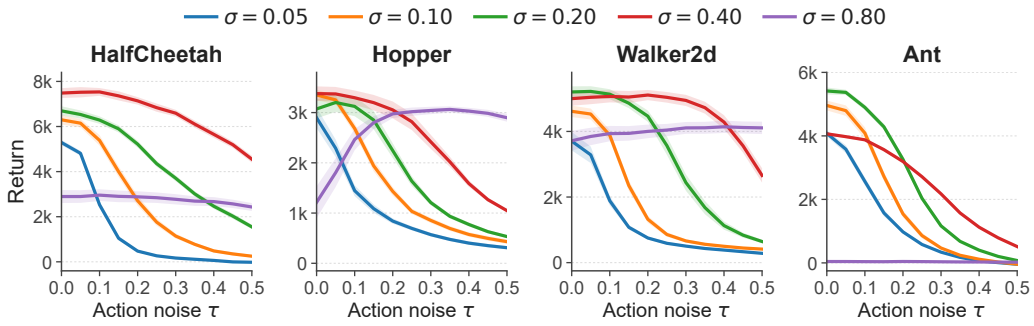

Figure 4: **Action-noise response across fixed training scales.** Mean $\pm$ SE over 15 seeds.

As the training scale increases from 0.05 to 0.20, action AUC improves monotonically across all four tasks. The complete evaluation curves further reveal that policies trained with larger scales sustain higher returns under heavier external action noise. When the scale is further raised to 0.40, this upward trend in action AUC persists in HalfCheetah, Hopper, and Walker2d; in Ant, action AUC peaks (1264 ± 15), but nominal zero-noise return begins to decline relative to $\sigma = 0.20$.

Table 8: **Fixed-scale PPO across preset training scales.** Bold marks the largest mean within each environment and metric.

Environment	Metric	$\sigma = 0.05$	$\sigma = 0.10$	$\sigma = 0.20$	$\sigma = 0.40$	$\sigma = 0.80$
HalfCheetah	Nominal Return	5203 ± 147	6269 ± 228	6721 ± 194	$\mathbf{7539 \pm 224}$	2939 ± 287
	Action AUC	605 ± 22	1301 ± 60	2181 ± 49	$\mathbf{3297 \pm 85}$	1392 ± 115
Hopper	Nominal Return	2912 ± 206	3323 ± 130	3095 ± 192	$\mathbf{3380 \pm 145}$	1195 ± 320
	Action AUC	488 ± 24	742 ± 25	920 ± 44	1262 ± 53	$\mathbf{1362 \pm 41}$
Walker2d	Nominal Return	3691 ± 286	4650 ± 162	$\mathbf{5213 \pm 169}$	5047 ± 202	3749 ± 290
	Action AUC	562 ± 33	881 ± 32	1613 ± 74	$\mathbf{2333 \pm 85}$	2008 ± 99
Ant	Nominal Return	4043 ± 106	4924 ± 183	$\mathbf{5424 \pm 94}$	4081 ± 33	41 ± 3
	Action AUC	594 ± 19	866 ± 38	1252 ± 24	$\mathbf{1264 \pm 15}$	18 ± 1

When the scale is further increased to 0.80, substantial degradation and cross-task discrepancies emerge. In all four benchmarks, nominal zero-noise returns fall markedly below those of moderate-scale policies ($\sigma = 0.20$ and 0.40). The full response curves show distinct profiles: in HalfCheetah, the response curve remains relatively flat but lies strictly below the $\sigma = 0.40$ curve across all noise levels; Walker2d exhibits a crossover, outperforming $\sigma = 0.40$ only under heavy noise ($\tau \geq 0.45$) while suffering lower returns elsewhere; Hopper exhibits an initial increase in return as noise is injected; and Ant fails across the entire evaluation range (nominal return drops to 41 ± 3). Training scale thus governs both the policy's baseline capability and its perturbation envelope, with excessive scales inducing task-dependent trade-offs. Performance of these fixed-scale policies under friction and mass variations is reported in Appendix F.2.

C.3 Training Scale and Loss-Sharpness Readings

This section supports the sharpness comparison in Section 3.1. For each final checkpoint, we fix the mean parameters ψ and estimate the largest eigenvalue λ_+ of the Hessian of the PPO loss with respect to ψ, using 24 Lanczos iterations. We report two readings. The native reading uses the checkpoint's own scale. The common-scale reading replaces ϕ with the element-wise median over the 45 final checkpoints of the same environment (3 methods $\times$ 15 seeds) and keeps ψ unchanged. The two readings differ only in the σ used by the loss; the deterministic mean policy is the same.

The median same-seed SAM-PPO/PPO ratio is 0.025–0.092 under the native-scale reading. Under the common-scale reading, the medians become 1.96, 0.58, 0.93, and 1.32 for HalfCheetah, Hopper, Walker2d, and Ant, respectively. The native-scale sharpness contrast is therefore highly sensitive to the policy scale used in the loss evaluation. RFD has the largest mean common-scale curvature in all four tasks.

Table 9: **Mean-block curvature under native and common scales.**

Environment	Method	λ_+ (native)	λ_+ (common scale)
HalfCheetah	PPO	2573.3 ± 318.8	137.4 ± 13.9
	SAM-PPO	247.3 ± 38.0	343.2 ± 54.1
	RFD-PPO	157.1 ± 27.3	769.9 ± 126.1
Hopper	PPO	1847.8 ± 307.2	130.2 ± 15.5
	SAM-PPO	60.9 ± 7.3	77.9 ± 11.3
	RFD-PPO	42.2 ± 5.2	1969.2 ± 357.6
Walker2d	PPO	1238.6 ± 102.8	84.9 ± 8.5
	SAM-PPO	59.6 ± 8.8	107.8 ± 21.2
	RFD-PPO	39.9 ± 4.9	448.9 ± 63.6
Ant	PPO	4105.3 ± 286.8	83.7 ± 9.6
	SAM-PPO	99.2 ± 8.1	104.2 ± 9.9
	RFD-PPO	83.0 ± 4.7	854.8 ± 53.9

D Scale Feedback and Explicit Control

D.1 Routing Interventions

The local expansion in Appendix A.1 derives the cross-feedback from mean perturbations to scale gradients, whose coordinate sum is non-negative, corresponding to a non-negative rate of change in differential entropy along the cross-feedback direction. To assess the empirical role of this channel during policy optimization, we systematically choose either the mean block ψ or the log-scale block ϕ as the perturbation source, and route the resulting additional gradient feedback to one of the two blocks, yielding four routing configurations. Throughout this section, arrows denote "perturbation source $\rightarrow$ additional feedback recipient". All routes follow the matched-dose perturbation and clean-norm clipping protocol defined in Appendix B.2 and Algorithm 2, with both mean and scale parameters remaining fully trainable.

Routing performance. Table 10 summarizes the final policy scale and behavioral performance under the four feedback routes. Blocks not receiving additional feedback retain their unperturbed gradients and are updated under the matched clean-norm control.

Table 10: **Feedback-routing outcomes.** Arrows denote source $\rightarrow$ recipient; parentheses report nAUC.

Environment	Routing	Policy scale	Nominal return	Action AUC (nAUC)
HalfCheetah	$\phi \rightarrow \phi$	0.040 ± 0.002	5148 ± 409	933 ± 91 (41.0%)
	$\phi \rightarrow \psi$	0.039 ± 0.001	5641 ± 318	1065 ± 53 (39.6%)
	$\psi \rightarrow \psi$	0.075 ± 0.002	5927 ± 408	1347 ± 132 (45.6%)
	$\psi \rightarrow \phi$	0.130 ± 0.003	$\mathbf{6791 \pm 312}$	$\mathbf{1669 \pm 129}$ (48.3%)
Hopper	$\phi \rightarrow \phi$	0.055 ± 0.003	3045 ± 93	488 ± 20 (32.1%)
	$\phi \rightarrow \psi$	0.058 ± 0.002	2639 ± 241	425 ± 27 (36.4%)
	$\psi \rightarrow \psi$	0.069 ± 0.002	3230 ± 79	537 ± 19 (33.7%)
	$\psi \rightarrow \phi$	0.194 ± 0.005	$\mathbf{3314 \pm 185}$	$\mathbf{789 \pm 44}$ (49.3%)
Walker2d	$\phi \rightarrow \phi$	0.069 ± 0.003	4596 ± 158	968 ± 74 (41.7%)
	$\phi \rightarrow \psi$	0.071 ± 0.003	4107 ± 119	771 ± 45 (36.1%)
	$\psi \rightarrow \psi$	0.110 ± 0.003	$\mathbf{5220 \pm 118}$	1125 ± 59 (43.2%)
	$\psi \rightarrow \phi$	0.233 ± 0.005	4257 ± 253	$\mathbf{1407 \pm 38}$ (71.6%)
Ant	$\phi \rightarrow \phi$	0.040 ± 0.001	4595 ± 176	780 ± 32 (33.8%)
	$\phi \rightarrow \psi$	0.042 ± 0.001	4247 ± 235	756 ± 43 (35.8%)
	$\psi \rightarrow \psi$	0.071 ± 0.001	5262 ± 128	909 ± 23 (34.8%)
	$\psi \rightarrow \phi$	0.185 ± 0.002	$\mathbf{5498 \pm 103}$	$\mathbf{1196 \pm 20}$ (44.5%)

The four routes show a consistent cross-environment grouping in final policy scale: the ϕ-sourced routes remain lowest, $\psi \rightarrow \psi$ is intermediate, and $\psi \rightarrow \phi$ reaches the highest scale in all four tasks, where it also attains the highest mean action AUC. Because scale parameters remain learnable throughout, the higher scale of $\psi \rightarrow \psi$ over the ϕ-sourced routes indicates that mean-block feedback can indirectly slow the natural decay of scale through the closed-loop training dynamics.

Holding the mean block fixed as the perturbation source and switching the additional gradient recipient from the mean block ($\psi \rightarrow \psi$) to the log-scale block ($\psi \rightarrow \phi$), action AUC exhibits consistent paired gains across all four benchmarks: $+321.4 \pm 119.5$ in HalfCheetah (95% CI $[+92.4, +542.8]$), $+252.3 \pm 39.7$ in Hopper ($[+178.3, +326.3]$), $+282.4 \pm 52.8$ in Walker2d ($[+178.8, +379.1]$), and $+287.0 \pm 27.3$ in Ant ($[+237.9, +340.2]$). Concurrently, changes in nominal returns exhibit task-dependent variation: in Walker2d, action AUC rises from 1125 to 1407 while nominal return drops from 5220 to 4257, highlighting a trade-off between nominal capability and perturbation tolerance. Under this routing intervention, $\psi \rightarrow \phi$ reaches the highest mean final scale and action AUC in all four tasks. This supports a functional role for mean-to-scale feedback during training. Performance of these policies under friction and mass variations is reported in Appendix F.2.

D.2 Explicit Scale Control and Entropy Calibration

The routing experiments in Appendix D.1 show that adversarial feedback from the mean block to the scale block sustains the training scale. We also regulate the policy scale explicitly using standard entropy regularization and RFD and evaluate the resulting action-noise robustness. The entropy coefficient is selected by minimizing the trajectory RMSE relative to same-seed SAM. RFD introduces an inverse-variance penalty with a one-update-lag schedule, as detailed in Appendix B.2 and Algorithm 3.

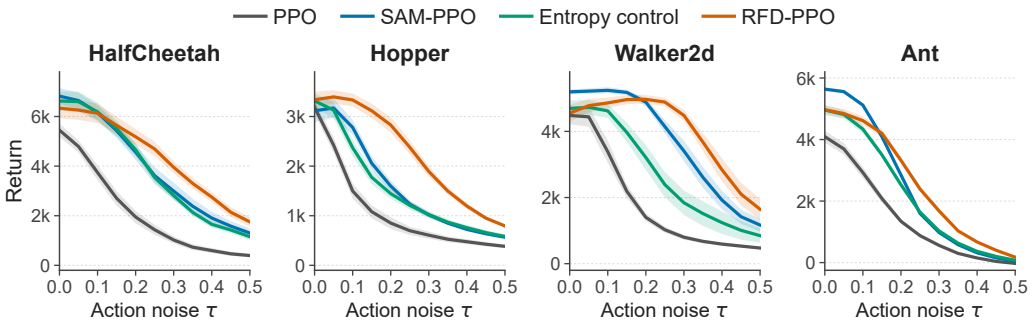

Figure 5: **Action-noise responses under native and explicit scale controls.** Mean ± SE; entropy uses trajectory-matched β^*.

Entropy calibration and performance. Table 11 reports the trajectory matching error alongside the behavioral performance of the calibrated entropy policies. We evaluate six candidate coefficients across 10 calibration seeds and select β^* by minimizing the mean trajectory RMSE relative to same-seed SAM. The candidate grid and calibration protocol are given in Appendix B.2. Behavioral evaluation is then expanded to 15 training seeds, including the 10 calibration seeds.

Table 11: **Entropy calibration and performance.** β^* minimizes scale-trajectory RMSE to same-seed SAM.

Environment	Selected β^*	Trajectory RMSE	Nominal return	Action AUC (nAUC)
HalfCheetah	0.0075	0.04218	6678 ± 385	1921 ± 94 (58.9%)
Hopper	0.0075	0.04481	3322 ± 144	759 ± 43 (46.0%)
Walker2d	0.005	0.05664	4686 ± 225	1365 ± 138 (57.6%)
Ant	0.0075	0.01897	4902 ± 154	1079 ± 28 (43.8%)

Across all four tasks, the selected entropy baseline achieves higher action AUC than PPO. In HalfCheetah and Hopper, its action AUC closely approaches native SAM, whereas gaps remain in Walker2d and Ant. Figure 5 shows where these differences emerge across the noise spectrum: the evaluation curves closely match in HalfCheetah, Hopper aligns with SAM under moderate-to-high noise, and Walker2d exhibits a steeper decline under intermediate noise. The calibrated entropy control recovers part of the action-noise improvement associated with larger training scales, but it does not reproduce SAM uniformly across tasks. In Ant, the final scales are similar (0.175 versus 0.168) while the action-AUC gap remains (1079 versus 1204); in Walker2d, entropy also finishes at a lower scale (0.227 versus 0.292).

Across all four tasks, RFD achieves higher mean action AUC than native SAM, with its response curves sustaining higher returns under moderate-to-high noise levels. At the same time, nominal returns for RFD in HalfCheetah, Walker2d, and Ant fall below those of SAM. Together, these two explicit controls demonstrate that directly regulating training scale produces substantial action-noise robustness; however, the exact robustness gains and their trade-offs with nominal performance de-

pend on both the control mechanism and task dynamics. Performance of these policies under friction and mass variations is reported in Appendix F.2.

E Endpoint Geometry Under Fixed Scale

E.1 Mean Updater, Parameter Dose, and Behavior

This section provides the complete action-noise evaluation curves supporting the fixed-scale comparison in Section 5 and Table 3. Both PPO and Mean-SAM fix the policy scale at $\sigma = 0.20$ throughout training and differ solely in the mean update rule; their nominal returns, action AUCs, and parameter-space diagnostics are summarized in Table 3.

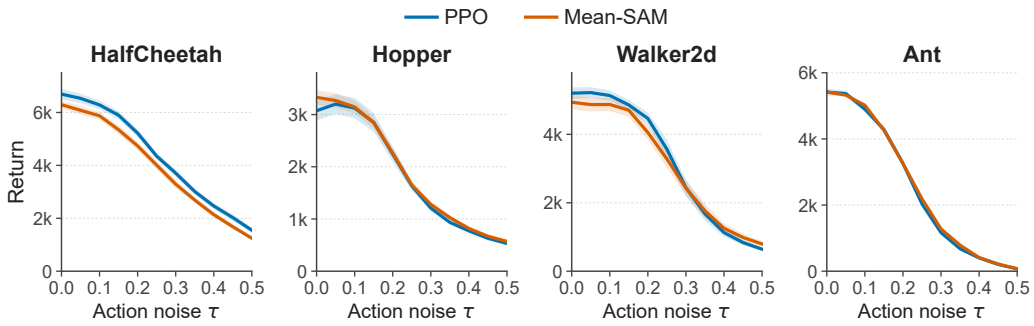

Figure 6: **Action-noise responses at fixed scale.** Mean ± SE at $\sigma = 0.20$.

As reported in Table 3, Mean-SAM achieves higher A_{80} estimates across all four benchmarks on relatively normalized parameter grids, but this does not translate into consistent action-noise robustness gains. The complete evaluation curves in Figure 6 reveal distinct task-specific behaviors: in HalfCheetah, Mean-SAM yields lower action AUC than PPO (1981 ± 51 versus 2181 ± 49); in Hopper (946 ± 34 versus 920 ± 44) and Ant (1275 ± 14 versus 1252 ± 24), the differences remain within sample error; and in Walker2d, the two response curves cross over across intermediate noise levels.

Concurrently, under relative normalization, Mean-SAM receives a smaller absolute parameter RMS dose D_{rms} (0.74–0.83$\times$ that of PPO across benchmarks) and produces smaller realized action displacement $\bar{d}$. Thus, the apparent flatness advantage on relative grids coincides with a smaller absolute parameter RMS dose and smaller realized action displacement, motivating the common-dose and displacement-matched analyses in Appendix E.2.

E.2 Common Action-Displacement Budget

This section details the common parameter RMS dose and action-displacement matching comparisons introduced in Section 5. Table 12 presents the three-step progression from relative grids to common RMS dose and one-to-one displacement matching. Common RMS evaluation retains each model's random directions while rescaling its grid radius to match the seed-matched PPO RMS dose. Subsequent displacement matching pairs noncentral points one to one without replacement within a 5% pooled IQR tolerance on shared reference states, following the protocol in Appendix B.3.

Table 12: **Return retention under dose and displacement controls.** Mean-SAM minus PPO at fixed $\sigma = 0.20$.

Environment	Relative grid ΔA_{80}	Common RMS ΔA_{80}	Matched Δp_{80}
HalfCheetah	$+0.091 \pm 0.043$	-0.017 ± 0.041	-0.032 ± 0.047
Hopper	$+0.080 \pm 0.057$	-0.052 ± 0.060	$+0.017 \pm 0.060$
Walker2d	$+0.100 \pm 0.057$	-0.050 ± 0.046	-0.020 ± 0.079
Ant	$+0.251 \pm 0.039$	$+0.133 \pm 0.044$	$+0.054 \pm 0.049$

Matching coverage and balance. Table 13 summarizes the coverage and displacement balance under the default 5% IQR tolerance. All matching operations are performed strictly based on action displacements without using return values.

Table 13: **Coverage and balance of displacement matching.** Gap metrics are normalized by the pooled IQR.

Environment	Mean matched pairs	Min matched pairs	Mean gap / IQR	W_1 / IQR
HalfCheetah	94.9	77	$+0.0006$	0.0158
Hopper	79.4	37	$+0.0001$	0.0133
Walker2d	76.5	39	-0.0006	0.0130
Ant	84.1	55	-0.0013	0.0145

Across environments, the matching achieves substantial overlap (averaging 76.5–94.9 pairs out of 120 candidates) while reducing the mean displacement gap and Wasserstein distance to negligible fractions of the IQR ($W_1/\text{IQR} \leq 0.016$). Comparisons are restricted to this shared support; scalar displacement matching does not constrain perturbation direction or state-level structure, and matched subsets no longer share identical parameter RMS doses.

Sensitivity to matching rules. Table 14 evaluates the robustness of the retention-fraction differences across three matching tolerances (2.5%, 5%, and 10% IQR) and three equal-frequency binning rules (5, 10, and 20 bins).

Table 14: **Sensitivity to displacement-matching rules.** Paired Δp_{80} is Mean-SAM minus PPO.

Rule	HalfCheetah	Hopper	Walker2d	Ant
2.5% IQR	-0.031 ± 0.046	$+0.009 \pm 0.061$	-0.024 ± 0.079	$+0.048 \pm 0.050$
5% IQR	-0.032 ± 0.047	$+0.017 \pm 0.060$	-0.020 ± 0.079	$+0.054 \pm 0.049$
10% IQR	-0.033 ± 0.048	$+0.023 \pm 0.060$	-0.022 ± 0.076	$+0.056 \pm 0.048$
5 bins	-0.036 ± 0.050	$+0.020 \pm 0.054$	-0.007 ± 0.075	$+0.053 \pm 0.047$
10 bins	-0.036 ± 0.050	$+0.020 \pm 0.056$	-0.011 ± 0.077	$+0.049 \pm 0.045$
20 bins	-0.038 ± 0.048	$+0.026 \pm 0.058$	-0.010 ± 0.077	$+0.044 \pm 0.045$

Across all six evaluated rules, the signs and magnitudes of the retention differences remain consistent: HalfCheetah and Walker2d point estimates remain negative, while Hopper and Ant remain moderately positive. None of the matched comparisons show a uniform cross-task retention advantage. These diagnostics show that the apparent flatness differences are sensitive to perturbation intensity. After matching the parameter RMS dose and further aligning realized action displacement, no cross-task-consistent return-retention advantage remains.

E.3 L2-Regularized Endpoint Geometry

To assess whether the decoupling between parameter-space flatness and action-noise robustness is unique to SAM or reflects a broader property of parameter-space regularization under fixed scales,

we evaluate standard L_2 regularization on the policy mean network. Training is conducted with policy scale held fixed at $\sigma = 0.20$ on HalfCheetah and Ant across three preset penalty coefficients $\lambda \in \{3 \times 10^{-5}, 10^{-4}, 3 \times 10^{-4}\}$. Table 15 reports behavioral performance alongside parameter-space diagnostics under relative grid normalization.

Table 15: L_2 **regularization at fixed scale** ($\sigma = 0.20$)**.** Gray rows are PPO baselines.

Environment	λ	Behavioral performance		Parameter-space diagnostics		
		Nominal return	Action AUC (nAUC)	A_{80}	$\bar{d}$	Dose / PPO
HalfCheetah	0 (PPO)	6721±194	2181±49 (65.7%)	0.461±0.031	0.356±0.010	1.000
	3×10^{-5}	6450±218	2081±37 (65.4%)	0.473±0.038	0.347±0.011	0.895
	10^{-4}	**7005±156**	**2272±77** (64.7%)	0.554±0.044	0.290±0.013	0.750
	3×10^{-4}	6689±197	2101±85 (63.0%)	0.589±0.037	0.298±0.012	0.573
Ant	0 (PPO)	5424±94	1252±24 (46.3%)	0.718±0.034	0.344±0.009	1.000
	3×10^{-5}	5387±106	1292±23 (47.7%)	0.797±0.029	0.310±0.009	0.925
	10^{-4}	5337±167	1274±41 (47.5%)	0.841±0.028	0.300±0.009	0.788
	3×10^{-4}	**5612±80**	**1323±23** (47.1%)	0.945±0.013	0.221±0.006	0.547

Across both environments, increasing L_2 regularization systematically increases apparent flatness on relative parameter grids (higher A_{80} point estimates), while concurrently decreasing absolute parameter RMS doses (D_{rms} reduced to 0.55–0.93× baseline) and realized action displacements. However, action AUC exhibits no uniform improvement: in HalfCheetah, $\lambda = 10^{-4}$ achieves slightly higher AUC while the strongest penalty $\lambda = 3 \times 10^{-4}$ deteriorates performance; in Ant, changes in AUC remain modest across all three penalty levels. The complete action-noise response curves are illustrated in Figure 7.

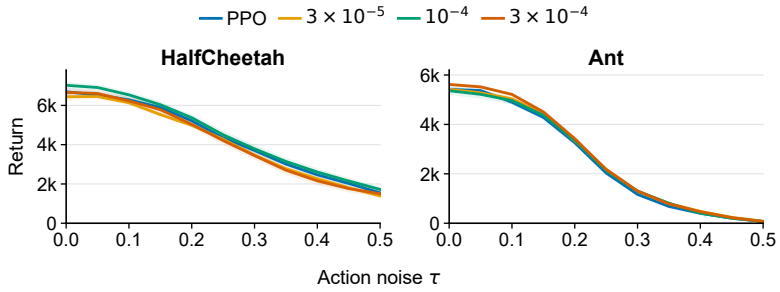

Figure 7: **Action-noise responses under L_2 regularization.** Mean ± SE at fixed $\sigma = 0.20$.

Common parameter dose and action-displacement matching. To eliminate the confounding effect of reduced perturbation doses, Table 16 evaluates these policies under common parameter RMS doses and 5% IQR action-displacement matching, following the evaluation protocols established in Appendix B.3 and Appendix E.2.

Once parameter doses are matched to unregularized PPO, five of the six A_{80} difference estimates become negative, with the weakest regularization setting in Ant remaining close to zero ($+0.008 \pm 0.040$). After equalizing parameter dose, the relative-grid flatness advantage is substantially reduced or reversed. Additional action-displacement matching likewise yields no consistent retention advantage across the evaluated settings; the strongest regularization in Ant remains negative (-0.106 ± 0.048).

Table 16: L_2 **regularization under matched dose and displacement.** Entries are L_2-PPO minus PPO.

Environment	λ	Common RMS ΔA_{80}	Matched Δp_{80}	Mean matched pairs	Min matched pairs
HalfCheetah	3×10^{-5}	-0.055 ± 0.043	-0.012 ± 0.062	94.8	64
	10^{-4}	-0.090 ± 0.040	-0.054 ± 0.042	92.4	72
	3×10^{-4}	-0.210 ± 0.036	-0.002 ± 0.056	70.5	43
Ant	3×10^{-5}	$+0.008 \pm 0.040$	-0.003 ± 0.037	94.5	73
	10^{-4}	-0.078 ± 0.043	$+0.014 \pm 0.055$	88.8	69
	3×10^{-4}	-0.217 ± 0.042	-0.106 ± 0.048	87.1	68

F Scope and Sensitivity

This appendix evaluates the boundaries of our empirical findings and provides sensitivity analyses. Appendix F.1 extends the investigation to two Box2D continuous-control tasks. Appendix F.2 examines robustness under physical parameter variations (friction and mass), alternative action-perturbation implementations, and observation noise, alongside a learnable-scale L_2-regularization baseline. Appendix F.3 evaluates the boundary between final and best online-return checkpoint selection. This appendix collects the scope and sensitivity checks of our study. Flatness is read on finite random 1D and 2D slices, the common action-displacement budget supports matched comparisons rather than causal mediation claims, and layer norms, absolute parameter doses, reference-state displacements, and action-noise robustness are four distinct quantities. Each analysis below reports its own quantity directly, and the conclusions stay within what the corresponding measurement supports.

F.1 Box2D Tasks

To evaluate whether the interplay among training scale, parameter-space geometry, and behavioral robustness extends beyond MuJoCo locomotion, we conduct parallel evaluations on two continuous Box2D benchmarks: `BipedalWalker-v3` and `LunarLanderContinuous-v3`. We compare native PPO and SAM, a five-level fixed-scale PPO sweep ($\sigma \in \{0.05, 0.10, 0.20, 0.40, 0.80\}$), and fixed-scale PPO versus Mean-SAM at $\sigma = 0.20$. Hyperparameters and evaluation protocols follow Appendices B.1–B.3, with 15 independent training seeds per condition.

Native algorithms and training-scale sweeps. Table 17 summarizes behavioral performance alongside parameter-space diagnostics under relative grid normalization. In both environments, native SAM converges to substantially higher final scales than PPO (0.434 versus 0.048 in BipedalWalker, and 0.158 versus 0.046 in LunarLander), accompanied by higher nominal returns and action AUCs. In BipedalWalker, mean action AUC increases from 96.2 ± 7.0 to 137.0 ± 2.5; in LunarLander, it rises from 83.9 ± 3.2 to 110.0 ± 2.7.

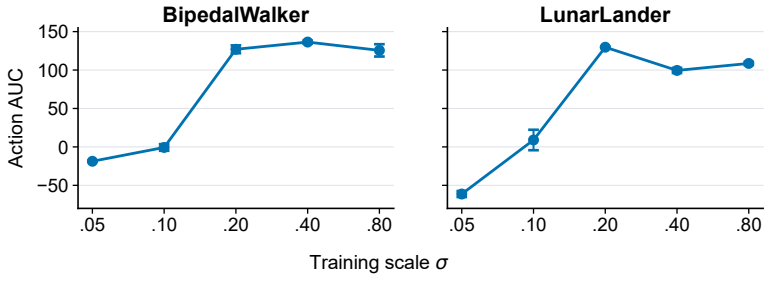

Figure 8: **Fixed-scale robustness in Box2D.** Mean ± SE over 15 seeds.

The five-level fixed-scale sweep in Figure 8 illustrates how robustness depends on the training-scale operating regime. In BipedalWalker, low scales ($\sigma \in \{0.05, 0.10\}$) yield poor nominal returns, whereas increasing scale to $\sigma = 0.20$ substantially improves action AUC to 127.0 ± 5.1. However, further scale expansion across $[0.20, 0.80]$ yields no additional monotonic gains. In LunarLander, both nominal return and action AUC peak at $\sigma = 0.20$, beyond which larger scales produce diminished returns.

Table 17: **Box2D native and fixed-scale comparisons.** Fixed-scale arms use $\sigma = 0.20$.

Training	Method	Policy scale	Behavioral performance		Parameter-space diagnostics	
			Nominal return	Action AUC	A_{80}	$\bar{d}$
BipedalWalker-v3						
Native	PPO	0.048±0.003	245.9±14.2	96.2±7.0	0.632±0.071	0.191±0.006
	SAM	0.434±0.020	**281.0±5.2**	**137.0±2.5**	0.884±0.042	0.173±0.014
Fixed σ	PPO	0.20	279.8±7.7	**127.0±5.1**	0.698±0.046	0.253±0.016
	Mean-SAM	0.20	**283.4±7.8**	106.2±6.0	0.666±0.038	0.143±0.009
LunarLanderContinuous-v3						
Native	PPO	0.046±0.003	206.3±6.2	83.9±3.2	0.993±0.003	0.091±0.007
	SAM	0.158±0.005	**256.5±5.0**	**110.0±2.7**	0.986±0.007	0.061±0.006
Fixed σ	PPO	0.20	**286.9±1.0**	**129.7±0.8**	0.984±0.010	0.098±0.010
	Mean-SAM	0.20	280.2±2.0	127.8±1.4	0.993±0.003	0.082±0.010

Fixed-scale mean updater comparisons. Holding the policy scale fixed at $\sigma = 0.20$, Mean-SAM produces smaller mean action displacements than PPO in BipedalWalker (0.143 ± 0.009 versus 0.253 ± 0.016), but fails to improve A_{80} (0.666 versus 0.698), and has a paired action-AUC deficit of -20.8 ± 7.8. In LunarLander, both PPO and Mean-SAM exhibit near-ceiling A_{80} retentions (> 0.98), while the paired action-AUC difference is small relative to its uncertainty (-1.9 ± 1.7 SE).

These Box2D results highlight two practical boundary conditions: first, robustness gains from scale expansion depend on the environment dynamics and the chosen scale range; second, parameter-space flatness metrics are subject to ceiling effects in simpler environments like LunarLander, where near-perfect return retention conceals behavioral differences across training conditions.

F.2 Performance Under Other Perturbations

This section evaluates the main algorithmic interventions across physical parameter variations (body out-of-distribution friction and mass), alternative action- and observation-noise implementations via Robust-Gymnasium, and a learnable-scale L_2-regularization baseline.

Physical parameter variations (Body OOD). To examine whether robustness gains extend to environmental dynamics shifts, we evaluate three test conditions against the nominal baseline: scaling designated contact and ground friction coefficients by 0.88, multiplying torso mass by 1.20, and their joint application. In MuJoCo, mass perturbation scales `model.body_mass` of the torso body by 1.20. Friction perturbation scales the tangential friction in `model.geom_friction` by 0.88 on designated distal contact geometries: `ffoot` in HalfCheetah, `foot` in Hopper, `foot_left` in Walker2d, and `left_ankle_geom` in Ant. Crucially, floor friction is scaled synchronously to 0.88 to prevent MuJoCo's default contact-friction combination rules from overriding distal friction changes. Each condition is evaluated over 50 deterministic-mean episodes without external action noise. Body OOD return is the unweighted arithmetic mean across the three perturbed conditions. Retention is normalized against zero-perturbation returns from the identical evaluation interface within each training seed before aggregation. Tables 18, 19, and 20 report performance across native methods, fixed scales, and routing interventions, respectively.

Table 18: **Body-OOD performance: native algorithms and scale replay.**

Panel / Environment	History → Updater	Body OOD Return [95% CI]	Retention [95% CI]
Native algorithms			
HalfCheetah	PPO	4774.3 [4222.7, 5404.5]	88.7% [81.8%, 94.9%]
	SAM	6610.2 [6053.8, 7231.6]	96.4% [93.9%, 99.3%]
	Entropy	6657.3 [5849.2, 7452.3]	99.0% [96.2%, 101.6%]
	RFD	6321.0 [5478.4, 7071.2]	99.4% [96.7%, 103.5%]
Hopper	PPO	861.6 [588.1, 1199.1]	27.7% [19.5%, 37.3%]
	SAM	1720.0 [1342.4, 2133.6]	62.3% [46.2%, 81.0%]
	Entropy	1805.1 [1477.1, 2152.1]	55.8% [46.3%, 66.5%]
	RFD	2916.3 [2558.3, 3252.0]	90.5% [78.0%, 103.8%]
Walker2d	PPO	3396.4 [2666.6, 4180.0]	72.4% [62.1%, 82.5%]
	SAM	5240.5 [5075.1, 5412.3]	101.0% [99.3%, 102.6%]
	Entropy	4376.7 [3785.5, 4954.5]	92.8% [86.2%, 98.4%]
	RFD	4815.7 [4392.5, 5199.3]	106.4% [99.7%, 115.7%]
Ant	PPO	4224.0 [3814.2, 4621.4]	105.3% [101.6%, 109.1%]
	SAM	5698.1 [5600.6, 5796.7]	100.4% [99.2%, 101.8%]
	Entropy	5022.3 [4713.9, 5305.4]	101.5% [100.2%, 102.9%]
	RFD	5049.1 [4753.7, 5288.6]	101.4% [100.3%, 102.6%]
Scale replay (2×2)			
HalfCheetah	PPO → PPO	5526.6 [4886.5, 6209.6]	91.3% [87.1%, 95.0%]
	SAM → PPO	6615.1 [5992.8, 7258.5]	97.8% [96.4%, 99.5%]
	PPO → Mean-SAM	4539.7 [3827.6, 5192.5]	90.0% [86.4%, 93.4%]
	SAM → Mean-SAM	6335.1 [5789.1, 6872.7]	99.6% [98.4%, 101.0%]
Hopper	PPO → PPO	967.9 [758.4, 1185.6]	33.8% [26.5%, 41.8%]
	SAM → PPO	2183.0 [1885.0, 2499.7]	75.3% [59.4%, 94.6%]
	PPO → Mean-SAM	1118.0 [973.8, 1270.8]	42.1% [34.8%, 50.6%]
	SAM → Mean-SAM	2075.8 [1727.3, 2418.6]	66.8% [53.2%, 81.8%]
Walker2d	PPO → PPO	2952.4 [2454.8, 3562.4]	64.3% [56.3%, 72.7%]
	SAM → PPO	4926.2 [4612.0, 5279.2]	103.2% [100.4%, 106.4%]
	PPO → Mean-SAM	3062.0 [2634.3, 3534.9]	67.6% [60.3%, 76.1%]
	SAM → Mean-SAM	4570.4 [4278.8, 4849.2]	107.7% [102.5%, 114.0%]
Ant	PPO → PPO	4797.2 [4487.9, 5081.4]	106.1% [103.7%, 108.7%]
	SAM → PPO	5499.7 [5330.2, 5662.6]	102.6% [100.5%, 104.9%]
	PPO → Mean-SAM	4975.1 [4710.9, 5211.9]	102.4% [100.9%, 104.0%]
	SAM → Mean-SAM	5752.3 [5665.9, 5830.6]	102.5% [101.7%, 103.4%]

Table 19: **Body-OOD performance across fixed PPO scales.**

Environment	Training Scale σ	Body OOD Return [95% CI]	Retention [95% CI]
HalfCheetah	0.05	5126.0 [4893.5, 5378.2]	97.6% [95.9%, 99.4%]
	0.10	6153.2 [5847.4, 6525.3]	99.1% [96.0%, 102.8%]
	0.20	6864.2 [6571.6, 7154.2]	102.9% [100.3%, 105.8%]
	0.40	7706.6 [7353.0, 8010.4]	101.1% [99.8%, 102.5%]
	0.80	3003.5 [2457.9, 3539.6]	103.8% [101.1%, 106.5%]
Hopper	0.05	1088.1 [918.1, 1277.7]	40.7% [33.0%, 49.8%]
	0.10	1774.8 [1403.0, 2139.5]	53.0% [43.1%, 62.1%]
	0.20	2575.8 [2172.2, 2956.6]	88.4% [70.5%, 107.7%]
	0.40	3271.8 [2981.9, 3473.3]	100.6% [88.1%, 117.1%]
	0.80	2948.2 [2597.2, 3216.3]	524.5% [341.6%, 716.2%]
Walker2d	0.05	2960.7 [2473.5, 3425.9]	81.7% [75.0%, 88.5%]
	0.10	4372.5 [3971.2, 4780.6]	93.9% [89.1%, 98.1%]
	0.20	5234.0 [4945.6, 5539.9]	100.8% [98.7%, 103.1%]
	0.40	5125.0 [4680.8, 5548.2]	102.0% [100.3%, 103.4%]
	0.80	3913.3 [3483.5, 4358.8]	112.1% [100.8%, 132.6%]
Ant	0.05	4035.2 [3845.0, 4233.4]	101.4% [100.3%, 102.6%]
	0.10	4974.1 [4652.9, 5308.0]	100.6% [99.7%, 101.5%]
	0.20	5514.9 [5362.2, 5657.1]	102.5% [101.7%, 103.4%]
	0.40	4137.9 [4039.8, 4232.0]	102.2% [101.1%, 103.2%]
	0.80	41.9 [36.3, 47.6]	94.9% [86.9%, 102.4%]

Table 20: **Body-OOD performance across feedback routes.**

Environment	Routing Configuration	Body OOD Return [95% CI]	Retention [95% CI]
HalfCheetah	$\phi \to \phi$	4462.6 [3745.4, 5045.8]	91.5% [84.8%, 99.2%]
	$\phi \to \psi$	5002.2 [4491.3, 5484.5]	89.8% [84.1%, 95.4%]
	$\psi \to \psi$	5251.8 [4345.1, 6156.1]	86.8% [79.4%, 92.9%]
	$\psi \to \phi$	6396.6 [5816.4, 7058.7]	94.6% [91.5%, 96.8%]
Hopper	$\phi \to \phi$	834.6 [714.4, 972.6]	27.8% [23.5%, 32.3%]
	$\phi \to \psi$	917.4 [690.0, 1169.6]	40.1% [29.4%, 51.7%]
	$\psi \to \psi$	999.3 [863.8, 1158.5]	31.1% [27.3%, 35.4%]
	$\psi \to \phi$	1933.7 [1570.7, 2336.9]	59.9% [49.1%, 71.6%]
Walker2d	$\phi \to \phi$	3498.8 [3030.7, 4024.8]	75.6% [68.6%, 83.0%]
	$\phi \to \psi$	2887.4 [2296.6, 3494.4]	68.9% [57.2%, 80.3%]
	$\psi \to \psi$	4394.0 [3869.7, 4873.1]	84.4% [75.1%, 92.9%]
	$\psi \to \phi$	4586.6 [4306.4, 4847.9]	120.2% [98.9%, 146.3%]
Ant	$\phi \to \phi$	4712.6 [4332.3, 5050.4]	101.5% [98.7%, 104.1%]
	$\phi \to \psi$	4350.7 [3904.7, 4766.9]	101.6% [99.5%, 103.8%]
	$\psi \to \psi$	5353.1 [5157.1, 5551.8]	103.1% [101.4%, 104.8%]
	$\psi \to \phi$	5561.3 [5342.5, 5770.6]	100.2% [99.0%, 101.3%]

Across HalfCheetah, Hopper, and Walker2d, replacing the PPO scale history with the SAM scale history systematically increases mean Body OOD retention under both mean update rules (e.g., under PPO updates, Hopper retention rises from 33.8% to 75.3%). Similarly, holding the mean perturbation fixed and routing feedback to the scale block ($\psi \to \phi$) elevates retention across these tasks (Hopper rising from 31.1% to 59.9%). In Ant, retention remains tightly bounded around 100% across conditions, reflecting nominal baseline differences rather than physical sensitivity. Retention values exceeding 100% occur when policies maintain higher mean returns under perturbed dynamics than their respective zero-noise baselines.

Action and observation perturbations (Robust-Gymnasium). We evaluate policies under Gaussian action and state perturbations using Robust-Gymnasium (Gu et al., 2025) across 300 final checkpoint evaluations. The three noise standard deviations per channel are selected by a calibration run on PPO alone, targeting cross-environment mean retentions of about 0.85, 0.55, and 0.30. Once selected, the grid is held fixed for all methods. We evaluate $\tau_{\text{action}} \in \{0.05, 0.12, 0.18\}$ and $\tau_{\text{state}} \in \{0.02, 0.05, 0.12\}$ over 50 episodes per level. Table 21 presents retention rates normalized against each policy's zero-perturbation return under the respective channel.

Table 21: **Robust-Gymnasium retention under action and observation noise.**

Environment	Method	Action noise retention (%)			Observation noise retention (%)		
		$\tau = 0.05$	$\tau = 0.12$	$\tau = 0.18$	$\tau = 0.02$	$\tau = 0.05$	$\tau = 0.12$
HalfCheetah	PPO	86.7 ± 3.3	55.7 ± 7.0	37.6 ± 6.3	78.8 ± 5.7	40.6 ± 7.5	12.1 ± 4.1
	SAM-PPO	96.5 ± 0.8	83.1 ± 2.4	66.0 ± 5.0	79.8 ± 3.3	45.6 ± 5.0	15.9 ± 2.9
	RFD-PPO	99.9 ± 1.5	92.5 ± 2.2	85.0 ± 2.5	80.6 ± 3.1	45.5 ± 5.4	16.5 ± 4.4
	Entropy β^*	97.7 ± 0.9	86.0 ± 2.0	71.4 ± 3.7	67.7 ± 5.1	34.3 ± 7.6	14.1 ± 4.7
	L2-PPO	85.0 ± 3.4	46.6 ± 6.8	28.0 ± 6.0	86.4 ± 5.5	53.8 ± 6.9	19.2 ± 5.4
Hopper	PPO	75.3 ± 6.2	42.2 ± 5.4	30.1 ± 4.0	50.0 ± 6.8	22.0 ± 3.5	8.0 ± 0.9
	SAM-PPO	105.1 ± 5.1	90.2 ± 12.6	64.6 ± 10.3	89.9 ± 4.8	58.8 ± 9.0	23.4 ± 4.6
	RFD-PPO	103.0 ± 2.7	100.1 ± 5.6	90.7 ± 6.2	101.5 ± 4.5	62.3 ± 7.1	12.9 ± 2.2
	Entropy β^*	94.5 ± 3.4	64.6 ± 4.3	47.0 ± 3.1	76.9 ± 4.4	32.3 ± 5.3	10.3 ± 2.4
	L2-PPO	81.9 ± 4.1	50.5 ± 4.2	38.4 ± 3.7	58.8 ± 5.1	30.0 ± 4.2	12.7 ± 2.2
Walker2d	PPO	98.1 ± 2.2	58.4 ± 3.3	34.0 ± 2.3	82.1 ± 5.1	43.3 ± 6.1	19.6 ± 2.7
	SAM-PPO	100.7 ± 0.5	100.8 ± 0.9	96.8 ± 1.4	97.2 ± 1.5	75.0 ± 6.5	28.1 ± 6.0
	RFD-PPO	105.4 ± 3.0	109.2 ± 4.9	111.6 ± 6.9	101.4 ± 4.0	67.4 ± 8.0	30.3 ± 8.5
	Entropy β^*	99.8 ± 1.6	96.4 ± 2.5	75.7 ± 5.5	90.3 ± 2.8	57.9 ± 7.4	17.5 ± 3.6
	L2-PPO	95.9 ± 1.5	61.2 ± 3.5	37.0 ± 2.8	77.1 ± 6.0	32.5 ± 4.7	12.7 ± 2.6
Ant	PPO	94.9 ± 1.9	77.9 ± 2.5	60.9 ± 2.8	95.6 ± 2.4	79.5 ± 3.2	47.1 ± 4.2
	SAM-PPO	98.3 ± 0.8	89.9 ± 1.1	73.5 ± 1.6	97.4 ± 0.8	84.1 ± 2.2	49.1 ± 3.9
	RFD-PPO	98.3 ± 0.6	92.7 ± 0.8	83.0 ± 1.2	96.7 ± 0.9	86.2 ± 1.8	50.8 ± 4.7
	Entropy β^*	96.3 ± 1.0	89.4 ± 1.3	75.3 ± 1.2	96.5 ± 1.0	80.2 ± 3.2	43.0 ± 4.1
	L2-PPO	95.1 ± 1.3	77.8 ± 2.3	55.6 ± 2.7	96.5 ± 1.4	79.2 ± 2.3	46.4 ± 4.5

Under the highest action-noise level, SAM achieves higher retention than PPO across all four benchmarks, with RFD exceeding SAM. Under observation noise, SAM also outperforms PPO across benchmarks, but the relative ordering between RFD and SAM depends on the task (e.g., in Hopper, SAM and RFD achieve 23.4% and 12.9%, whereas in Walker2d they attain 28.1% and 30.3%). Explicit scale control yields consistent retention benefits under action noise, while observation-noise robustness exhibits pronounced environment-specific behavior.

Learnable-scale L_2 regularization baseline. We add an L_2 penalty ($\lambda = 10^{-4}$) to the PPO mean network while keeping policy scales learnable. Table 22 reports final scales, nominal returns, and action AUC; Table 21 gives the cross-method Robust-Gymnasium comparison.

Table 22: **Learnable-scale L_2 regularization.**

Environment	PPO Final Scale	L2 Final Scale	L2 Nominal Return	L2 Action AUC
HalfCheetah	0.0410 ± 0.0012	0.0484 ± 0.0011	5296.4 ± 206.6	941.3 ± 64.9
Hopper	0.0560 ± 0.0022	0.0663 ± 0.0031	2900.8 ± 178.1	550.4 ± 28.5
Walker2d	0.0698 ± 0.0031	0.0838 ± 0.0024	4484.8 ± 225.6	864.9 ± 43.5
Ant	0.0388 ± 0.0011	0.0473 ± 0.0011	4715.1 ± 171.1	809.8 ± 34.8

Final scales are only 18%–22% above PPO. In Table 21, L2-PPO remains below SAM-PPO at the strongest action perturbation in all four benchmarks.

F.3 Checkpoint and Measurement Boundaries

The main text strictly evaluates terminal checkpoints at the completion of 3M training steps. To assess sensitivity to checkpoint selection, we additionally evaluate `best` checkpoints, chosen by the highest 10-episode online nominal-return estimate during training and subsequently evaluated post hoc across 50 independent test episodes with a distinct episode-seed stream under our 11-level action-noise protocol. Table 23 reports the paired difference in action AUC between fixed-scale Mean-SAM ($\sigma = 0.20$) and standard PPO ($\sigma = 0.20$) across both checkpoint criteria.

Table 23: **Checkpoint sensitivity of the fixed-scale mean-updater comparison.**

Environment	Final: ΔAction AUC	Best: ΔAction AUC
HalfCheetah	-200.0 ± 73.4	-231.1 ± 79.8
Hopper	25.5 ± 50.4	-25.4 ± 47.4
Walker2d	-58.5 ± 113.2	-141.6 ± 125.1
Ant	22.4 ± 26.3	-0.1 ± 31.2

Under neither checkpoint criterion does the mean updater provide a cross-task-consistent action-noise robustness benefit over standard PPO. In Hopper and Ant, the paired differences are small relative to their uncertainty under both checkpoint criteria, and their point estimates change sign. In Ant at $\sigma = 0.80$, the final checkpoint attains an action AUC of only 17.8 ± 1.1, whereas the best online-return checkpoint reaches 270.8 ± 2.9, showing substantial sensitivity to checkpoint selection in this extreme-scale setting. All geometric comparisons in Section 5 use the terminal 3M checkpoint.