VPPO: VERIFIED PATHWISE POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Proximal policy optimization (PPO) reuses on-policy data through a clipped surrogate, but clipping alone does not explicitly bound policy change. Incorporating critic action gradients offers a complementary direction for policy improvement, provided that their reliability can be assessed. We introduce Verified Pathwise Policy Optimization (VPPO), an on-policy algorithm that combines clipped natural-gradient updates with probe-calibrated pathwise refinement. More specifically, VPPO performs multiple re-linearized steps on each rollout batch, subject to explicit KL bounds and nondecrease of the regularized surrogate. To calibrate trust in the critic-guided refinement, we use paired simulator trajectories with common random numbers to evaluate the proposed action direction. A damped-Fisher projection then preserves first-order compatibility between the refinement and surrogate directions. A matched-budget ablation shows that clipping remains necessary for stable batch reuse even under explicit KL control. On GPU-parallelized benchmarks, VPPO improves humanoid locomotion performance over the standard PPO configuration while achieving comparable final returns on additional control tasks. Under matched entropy regularization, VPPO reaches intermediate performance milestones earlier than PPO. Compared with off-policy baselines, VPPO also reaches target returns in less wall-clock time on most tested control tasks, despite equiring more environment steps.

1 INTRODUCTION

Proximal policy optimization (PPO) (Schulman et al., 2017) is widely used in the field of reinforcement learning, including for massively parallel training of legged locomotion (Rudin et al., 2022; Schwarke et al., 2025). Conventionally, in the PPO algorithm, each rollout batch is reused over several epochs by Minibatch-Adam on a truncated surrogate. At the same time, the learning rate is adjusted in response to the observed KL divergence (Ilyas et al., 2020; Engstrom et al., 2020; Andrychowicz et al., 2021). These mechanisms collectively shape the update, but clipping does not directly constrain the realized policy change, and the learning rate adjustment only affects subsequent steps. We therefore separate batch reuse from the control of policy change: clipping remains within the surrogate, each proposed update is verified against empirical KL bounds, and the contribution of the critic's action gradients is calibrated using directional evidence.

In this paper, we introduce the Verified Pathwise Policy Optimization (VPPO) algorithm (see Figure 1). This on-policy optimizer separates batch reuse from policy-update control and adds a probe-calibrated refinement to the critic gradient. VPPO retains PPO's clipped surrogate but replaces the Adam and KL adaptive learning-rate updates over multiple epochs with up to K re-linearized natural-gradient steps (Schulman et al., 2015). At each step, it combines the surrogate gradient direction with a projected refinement of the critic gradient, then scales the result to a given KL budget. A candidate is accepted only if it complies with the empirical, stepwise, and rollout-relative KL bounds without reducing the regularized surrogate value. If the combined direction fails, VPPO tries again using only the surrogate gradient direction. VPPO extends the direction of the surrogate gradient by refining the action value gradient, whose impact must account for the critic's error (Tucker et al., 2018). Paired simulator branches that use shared random numbers (Glasserman, 2004) perturb an action both along and against the direction suggested by the critic. The resulting difference in the truncated, bootstrap-estimated sample result provides a directional cue; information excluded from the critic training is used to calibrate a confidence weight based on disagreements within the critic ensemble. More specifically, VPPO first removes components of the refinement that oppose the score direction

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.

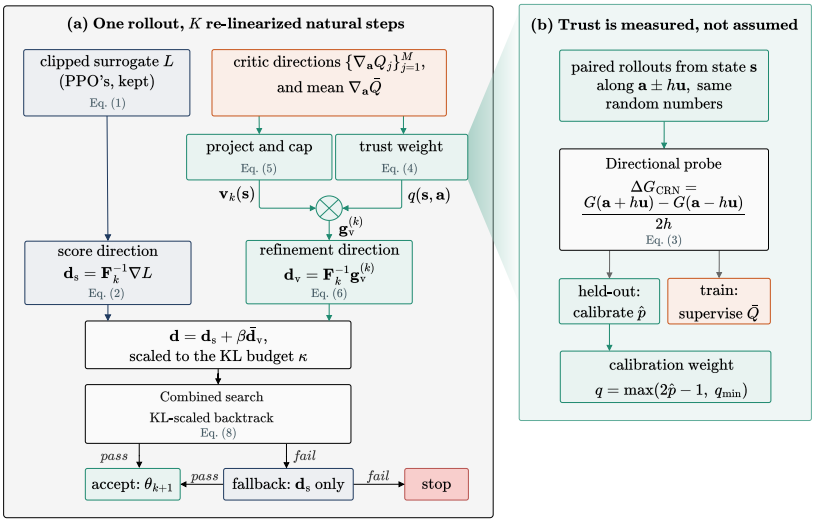


Figure 1: VPPO combines a clipped natural-gradient direction with probe-calibrated refinement. Action-space projection and the cap δ/K control per-state refinement targets. Damped-Fisher projection aligns the aggregated refinement with the surrogate direction. Paired trajectories with shared randomness provide directional labels, and held-out labels calibrate trust q. Each update must satisfy empirical stepwise and rollout-relative KL bounds without decreasing the regularized surrogate.

determined in each state, and then projects the aggregated refinement onto the damped Fisher metric. For an exact Fisher solution, the resulting combined direction is an ascent direction for the first-order regularized surrogate.

Our contributions are:

1. We develop a clipped natural-gradient updater that reuses each rollout through up to K re-linearized steps, accepting each step only after empirical per-step and rollout-relative KL and surrogate checks (Sections 4 and 7.1).
2. We develop a probe-calibrated pathwise refinement whose influence depends on directional evidence and whose action-space and damped-Fisher projections preserve first-order compatibility under an exact Fisher solve (Sections 4 and 5).
3. We use matched-budget ablations to examine clipping and repeated batch reuse, analyze the effect of entropy on the natural-gradient update, and evaluate learning, curriculum progress, and computational cost against on-policy baselines, with FastTD3 and FastSAC as off-policy references (Sections 6 and 7, Appendix G).

2 Related Work

Natural policy gradients use the Fisher metric to choose a policy update direction (Amari, 1998; Kakade, 2001; Peters & Schaal, 2008). TRPO uses this geometry to limit how far the policy moves in KL divergence (Schulman et al., 2015), building on work that relates small policy changes to performance improvement (Kakade & Langford, 2002; Achiam et al., 2017). PPO's clipped variant is easier to optimize, but clipping does not itself bound the KL divergence of the resulting policy (Schulman et al., 2017; Ilyas et al., 2020; Wang et al., 2020). Its behavior also depends strongly on implementation choices (Engstrom et al., 2020; Huang et al., 2022; Andrychowicz et al., 2021; Hsu et al., 2020). Phasic Policy Gradient separates policy and value-function training to make better use of collected data (Cobbe et al., 2021). VPPO separates clipping for batch reuse from explicit KL constraints on policy change.

Critic gradients offer another way to improve policy updates. Action-dependent control variates can incorporate them while retaining an unbiased policy-gradient estimator under their assumptions (Gu et al., 2017; Liu et al., 2018). Their benefit is not automatic: Tucker et al. (2018) found little variance reduction on their benchmarks and traced some earlier reported gains to biased implementations.

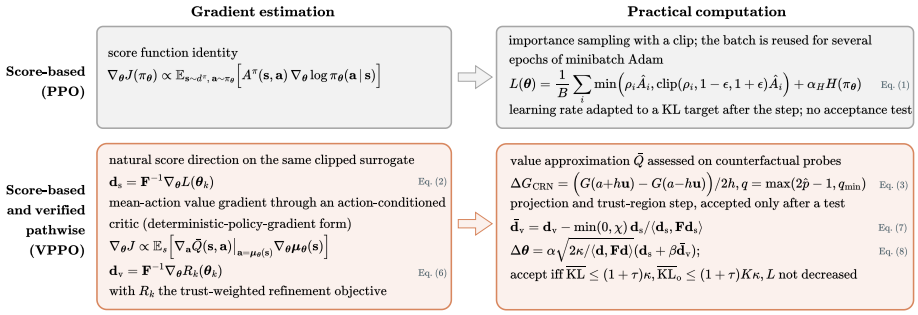


Figure 2: The gradient estimators used by PPO and VPPO and the steps that make them computable in practice. VPPO keeps PPO's clipped estimator in natural coordinates and adds a pathwise estimator whose directional reliability is measured on held-out probes before use.

Other methods use action-value gradients more directly (Silver et al., 2014; Heess et al., 2015; Lillicrap et al., 2016; Fujimoto et al., 2018; Haarnoja et al., 2018a). Among on-policy methods, Gurumurthy et al. (Gurumurthy et al., 2023) use action gradients from a learned critic to update the policy in massively parallel simulation. REPPO (Voelcker et al., 2026) also uses critic gradients in its policy update. VPPO uses simulator probes to assess the critic's proposed action direction and calibrate how strongly that direction influences the update.

Additionally, VPPO compares paired simulator branches under the same random numbers to reduce noise in this directional test, drawing on common-random-number methods in simulation (Glasserman, 2004) and fixed-seed policy search (Ng & Jordan, 2000). We also evaluate FastTD3 as an off-policy reference for environment-step efficiency (Seo et al., 2025). Its training regime differs from the on-policy methods, so we report it as context rather than a matched competitor (Appendix G).

Note that automatic entropy tuning is common in off-policy learning (Haarnoja et al., 2018b). In this work, we analyze how a fixed entropy coefficient affects policy noise under VPPO's natural-gradient update. We also report curriculum level alongside reward: when terrain difficulty changes during training, reward alone can give a misleading picture of progress, a concern related to the broader literature on reward specification (Amodei et al., 2016; Skalse et al., 2022).

3 PPO BATCH REUSE AND EXPLICIT TRUST REGIONS

Setting. We consider a discounted MDP with state space $\mathcal{S}$, action space $\mathcal{A} \subseteq \mathbb{R}^{n_A}$, discount $\gamma \in (0,1)$, return $J(\boldsymbol{\theta}) = \mathbb{E}\left[\sum_{t\ge 0}\gamma^t r_t\right]$, normalized discounted state visitation d^π, and advantage $A^\pi = Q^\pi - V^\pi$ (Sutton & Barto, 2018). The policy is a diagonal Gaussian $\pi_{\boldsymbol{\theta}}(\mathbf{a}\,|\,\mathbf{s}) = \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{s}), \mathrm{diag}(\boldsymbol{\sigma}^2)\right)$ (the environment may clip executed actions and the notation refers to the Gaussian action) with a state-independent $\boldsymbol{\psi} = \log\boldsymbol{\sigma} \in \mathbb{R}^{n_A}$ as part of $\boldsymbol{\theta}$, as in the locomotion library we build on (Schwarke et al., 2025). We denote $\pi_{\mathrm{o}} = \pi_{\boldsymbol{\theta}_{\mathrm{o}}}$ for the policy that collected the current batch, $\rho_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}) = \pi_{\boldsymbol{\theta}}(\mathbf{a}\,|\,\mathbf{s})/\pi_{\mathrm{o}}(\mathbf{a}\,|\,\mathbf{s})$ for the importance ratio, $\boldsymbol{\xi} = (\mathbf{a} - \boldsymbol{\mu}_{\mathrm{o}}(\mathbf{s}))/\boldsymbol{\sigma}_{\mathrm{o}}$ for the standardized noise of a sampled action, and $\hat{A}$ for the generalized advantage estimate (Schulman et al., 2016), normalized to zero mean and unit variance over the batch as in the reference implementation.

PPO's update. Given a batch $\mathcal{D} = \{(s_i, a_i, \hat{A}_i)\}_{i=1}^B$ collected by π_{o}, PPO maximizes the regularized clipped objective,

$$L(\boldsymbol{\theta}) = L^{\mathrm{clip}}(\boldsymbol{\theta}) + \alpha_H H(\pi_{\boldsymbol{\theta}}), \qquad L^{\mathrm{clip}}(\boldsymbol{\theta}) = \frac{1}{B}\sum_{i=1}^{B} \min\Big(\rho_i \hat{A}_i, \mathrm{clip}\left(\rho_i, 1-\epsilon, 1+\epsilon\right)\hat{A}_i\Big), \quad (1)$$

where $\rho_i = \rho_{\boldsymbol{\theta}}(\mathbf{s}_i, \mathbf{a}_i)$ and $H(\pi_{\boldsymbol{\theta}}) = \sum_j \left(\psi_j + \frac{1}{2}\log(2\pi e)\right)$. PPO uses several epochs of minibatch Adam (Kingma & Ba, 2015) on the same batch, with a learning rate raised or lowered according to whether the measured KL divergence to π_{o} falls below or above a target. The clipped surrogate is a pessimistic version of the importance-sampled surrogate $L^{\mathrm{IS}}(\boldsymbol{\theta}) = \frac{1}{B}\sum_i \rho_i \hat{A}_i$: the two coincide at $\boldsymbol{\theta}_{\mathrm{o}}$, and $L^{\mathrm{clip}} \le L^{\mathrm{IS}}$ pointwise.

Why retain the clipped surrogate? Figure 2 compares the PPO and VPPO updates. In the reference PPO implementation, measured KL divergence adjusts subsequent learning rates without rejecting the current update. The following lemma characterizes how clipping bounds the surrogate.

Lemma 1 (The clipped surrogate is bounded; the importance-sampled surrogate is not)**.** *Let the batch advantages have a zero mean. Then for every $\boldsymbol{\theta}$, $L^{\mathrm{clip}}(\boldsymbol{\theta}) \le \frac{\epsilon}{B}\sum_{i=1}^{B}|\hat{A}_i|$, and a sample contributes zero gradient to L^{clip} as soon as its ratio has moved beyond the clip boundary in the direction favoured by the sign of its advantage. If the mean network can drive $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{s}_i)$ to $\mathbf{a}_i$ for a sample with $\hat{A}_i > 0$ while missing the other sampled actions, then $\sup_{\boldsymbol{\theta}} L^{\mathrm{IS}}(\boldsymbol{\theta}) = +\infty$, attained by shrinking $\boldsymbol{\sigma}$.*

The proof is in Appendix B.1. Re-maximizing L^{IS} on a fixed batch rewards concentrating probability mass on sampled actions with positive advantage. A finite KL cap rules out the lemma's limiting sequence, since the divergence blows up as $\boldsymbol{\sigma} \to \mathbf{0}$, but leaves large finite per-sample incentives, because a few samples can absorb most of an average budget. Lemma 1 is a statement about the surrogate; the interpretation that clipping contributes to stable batch reuse, PPO's original motivation (Schulman et al., 2017), is tested empirically in the matched-budget experiment of Section 7.1, where removing the clip under an intact KL constraint substantially reduced performance. Accordingly, VPPO retains clipping as its batch-reuse mechanism.

4 Verified Pathwise Policy Optimization

Given a batch $\mathcal{D}$ collected by π_{o}, VPPO performs up to K updates from $\boldsymbol{\theta}_0 = \boldsymbol{\theta}_{\mathrm{o}}$. Each combines a clipped natural-gradient direction with probe-calibrated refinement through damped-Fisher projection, followed by line search enforcing empirical KL bounds and surrogate nondecrease (Figure 1).

At inner step k, VPPO optimizes the regularized clipped surrogate L in Eq. (1). The importance ratios retain π_{o} as their reference, while the gradient and local curvature are recomputed at the current parameters $\boldsymbol{\theta}_k$. Let $\mathbf{g}_{\mathrm{s}}^{(k)} = \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}_k)$. We define the damped Fisher matrix using the KL divergence averaged over rollout states:

$$\mathbf{F}_k \;=\; \nabla^2_{\boldsymbol{\theta}'}\,\overline{\mathrm{KL}}\big(\pi_{\boldsymbol{\theta}_k} \,\|\, \pi_{\boldsymbol{\theta}'}\big)\Big|_{\boldsymbol{\theta}'=\boldsymbol{\theta}_k} + \lambda \mathbf{I}, \qquad \overline{\mathrm{KL}}(\pi \,\|\, \pi') = \frac{1}{B}\sum_{i=1}^{B} \mathrm{KL}\big(\pi(\cdot \,|\, \mathbf{s}_i) \,\|\, \pi'(\cdot \,|\, \mathbf{s}_i)\big), \quad (2)$$

Here, $\lambda > 0$ makes $\mathbf{F}_k$ positive definite. The score direction $\mathbf{d}_{\mathrm{s}} = \mathbf{F}_k^{-1}\mathbf{g}_{\mathrm{s}}^{(k)}$ is approximated by conjugate gradients using Fisher-vector products on a fixed batch subsample (Schulman et al., 2015; Nocedal & Wright, 2006). Recomputing this direction after each accepted update adapts the local geometry to the current policy while retaining PPO's repeated use of the rollout batch.

4.1 Probe-Calibrated Pathwise Refinement

Critic and probes. VPPO learns an ensemble of M action-conditioned critics $\{Q_j\}_{j=1}^{M}$ using n-step SARSA regression with Polyak-averaged target networks. Let $\bar{Q}$ denote the ensemble mean value estimate. To assess its direction of action gradient, we initialize two simulator trajectories from the same state $\mathbf{s}$ and couple their randomness using common random numbers (Glasserman, 2004). At the probe action $\mathbf{a} = \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{s})$, the normalized critic direction is $\mathbf{u} = \nabla_{\mathbf{a}}\bar{Q}(\mathbf{s},\mathbf{a}) / \big\|\nabla_{\mathbf{a}}\bar{Q}(\mathbf{s},\mathbf{a})\big\|$. The two trajectories take initial actions $\mathbf{a} \pm h\mathbf{u}$ and then follow the mean policy, yielding the directional label

$$\Delta G_{\mathrm{CRN}} \;=\; \frac{G(a + h\mathbf{u}) - G(a - h\mathbf{u})}{2h}, \qquad G(\mathbf{a}') = \sum_{t=0}^{T_p - 1} \gamma^t\, r_t + \gamma^{T_p}\, \bar{Q}\big(\mathbf{s}_{T_p}, \boldsymbol{\mu}(\mathbf{s}_{T_p})\big), \quad (3)$$

where T_p is the probe horizon and $h > 0$ the perturbation magnitude. The central difference estimates a directional derivative of the truncated return with a critic-bootstrapped terminal value. Shared randomness isolates the effect of the initial action perturbation. A probe is retained only when both perturbed actions lie within the feasible action box, preserving the symmetry of the difference. Probe labels are assigned to two disjoint streams. The training stream augments critic regression with the directional loss $\mathrm{Huber}\big(\mathbf{u}^{\top}\nabla_{\mathbf{a}} Q_j(\mathbf{s},\mathbf{a}),\, \Delta G_{\mathrm{CRN}}\big)$. The calibration stream is excluded from critic training and estimates the trust weights defined next. Implementation details are provided in Appendix D.

Calibrated trust. Trust calibration uses disagreement among ensemble action gradients, measured by $U(\mathbf{s},\mathbf{a}) = \operatorname{tr}\operatorname{Cov}_j\left[\nabla_{\mathbf{a}} Q_j(\mathbf{s},\mathbf{a})\right] / \left\|\frac{1}{M}\sum_j \nabla_{\mathbf{a}} Q_j(\mathbf{s},\mathbf{a})\right\|^2$. We partition U into quantile bins and estimate the frequency $\hat{p}(U)$ of positive calibration labels $\Delta G_{\mathrm{CRN}} > 0$, indicating that the positive perturbation yields a higher probe return. The corresponding trust weight is

$$q(\mathbf{s},\mathbf{a}) \;=\; \max\left(2\,\hat{p}(U(\mathbf{s},\mathbf{a})) - 1,\; q_{\min}\right) \in (0,1]. \tag{4}$$

The weight increases with directional agreement and retains a minimum contribution through $q_{\min} > 0$. It reflects agreement on the probe distribution, with Proposition 3 characterizing the unfloored probability margin under fixed-bin assumptions. The refinement uses this weight to modulate the contribution of each rollout sample.

Per-state direction. At each inner step, VPPO evaluates the critic gradient at the current policy mean and scales it by the rollout standard deviation $\boldsymbol{\sigma}_{\mathrm{o}}$. The resulting action-space vector is projected onto a half-space aligned with the sampled score direction and capped in norm:

$$\begin{aligned} \mathbf{v}_k(\mathbf{s}) &= \mathrm{cap}_{\delta/K}\left(\Pi_{\mathrm{safe}}\left(\boldsymbol{\sigma}_{\mathrm{o}} \odot \nabla_{\mathbf{a}} \bar{Q}(\mathbf{s},\mathbf{a})\big|_{\mathbf{a}=\boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{s})}\right)\right), \\ \Pi_{\mathrm{safe}}(\mathbf{u}) &= \mathbf{u} - \min\left(0, \langle \mathbf{u},\, \mathbf{w}\rangle\right)\frac{\mathbf{w}}{\|\mathbf{w}\|^2}, \qquad \mathrm{cap}_{\varrho}(\mathbf{u}) = \mathbf{u}\,\min\!\left(1, \frac{\varrho}{\|\mathbf{u}\|}\right), \end{aligned} \tag{5}$$

where $\mathbf{w} = \hat{A}\,\boldsymbol{\xi}$ is the sampled score direction. The projection Π_{safe} removes the component opposing $\mathbf{w}$, and the cap bounds each scaled target by δ/K, giving a cumulative target budget of δ independently of K (Lemma 2). The KL constraints separately control policy displacement. Critic gradients are recomputed at the current policy means, while score vectors and trust weights remain tied to sampled rollout actions. Holding $\mathbf{v}_k$ and q fixed during differentiation, we define the refinement objective and its natural direction:

$$R_k(\boldsymbol{\theta}) \;=\; \frac{1}{B}\sum_{i=1}^{B} q(\mathbf{s}_i,\mathbf{a}_i)\left\langle \mathbf{v}_k(\mathbf{s}_i),\, \frac{\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{s}_i)}{\boldsymbol{\sigma}_{\mathrm{o}}}\right\rangle, \qquad \mathbf{g}_{\mathrm{v}}^{(k)} = \nabla_{\boldsymbol{\theta}} R_k(\boldsymbol{\theta}_k), \qquad \mathbf{d}_{\mathrm{v}} = \mathbf{F}_k^{-1}\mathbf{g}_{\mathrm{v}}^{(k)}. \tag{6}$$

The refinement propagates the critic gradient through the policy mean (Silver et al., 2014), without averaging over reparameterized actions (Heess et al., 2015). It has no direct log-scale gradient, but can affect policy scale indirectly through Fisher projection and joint KL scaling.

4.2 Damped-Fisher direction fusion

Alignment in action space need not be preserved by the network Jacobian, batch aggregation, and Fisher preconditioning (Section 7.2). We therefore project $\mathbf{d}_{\mathrm{v}}$ in the same damped-Fisher metric used to construct the update. For $\mathbf{d}_{\mathrm{s}} \neq 0$, let $\chi = \langle \mathbf{d}_{\mathrm{s}},\, \mathbf{F}_k\mathbf{d}_{\mathrm{v}}\rangle$ and define

$$\bar{\mathbf{d}}_{\mathrm{v}} \;=\; \mathbf{d}_{\mathrm{v}} - \min(0,\chi)\,\frac{\mathbf{d}_{\mathrm{s}}}{\langle \mathbf{d}_{\mathrm{s}},\, \mathbf{F}_k\mathbf{d}_{\mathrm{s}}\rangle}, \qquad \text{so that} \qquad \left\langle \mathbf{d}_{\mathrm{s}},\, \mathbf{F}_k\bar{\mathbf{d}}_{\mathrm{v}}\right\rangle = \max(0,\chi) \;\geq\; 0, \tag{7}$$

The combined direction is $\mathbf{d} = \mathbf{d}_{\mathrm{s}} + \beta\bar{\mathbf{d}}_{\mathrm{v}}$, where $\beta \geq 0$ is a fixed refinement weight. This projection gives the nearest direction to $\mathbf{d}_{\mathrm{v}}$ in the damped-Fisher metric subject to nonnegative alignment with $\mathbf{d}_{\mathrm{s}}$. With an exact Fisher solve, the combined direction is an ascent direction for L to first order for every $\beta \geq 0$ (Proposition 1).

4.3 KL-constrained acceptance

VPPO normalizes the combined direction to a per-step KL budget κ and selects a step length by backtracking over $\alpha \in \{1, \zeta, \zeta^2, \ldots\}$, with $0 < \zeta < 1$. Each candidate must satisfy two KL constraints and a surrogate improvement condition:

$$\Delta\boldsymbol{\theta}_k = \alpha\sqrt{\frac{2\kappa}{\langle \mathbf{d},\, \mathbf{F}_k\mathbf{d}\rangle}}\,\mathbf{d}, \qquad \text{accept iff} \quad \begin{cases} \overline{\mathrm{KL}}\left(\pi_{\boldsymbol{\theta}_k} \,\|\, \pi_{\boldsymbol{\theta}_k+\Delta\boldsymbol{\theta}_k}\right) \leq (1+\tau)\,\kappa, \\ \overline{\mathrm{KL}}\left(\pi_{\mathrm{o}} \,\|\, \pi_{\boldsymbol{\theta}_k+\Delta\boldsymbol{\theta}_k}\right) \leq (1+\tau)\,K\kappa, \\ L(\boldsymbol{\theta}_k + \Delta\boldsymbol{\theta}_k) \geq L(\boldsymbol{\theta}_k), \end{cases} \tag{8}$$

Here, τ is the relative KL tolerance. All constraints are evaluated on the full rollout batch. If backtracking fails for the combined direction, VPPO retries along $\mathbf{d}_{\mathrm{s}}$. If both searches fail or $\mathbf{d}_{\mathrm{s}} = 0$, the batch update terminates at the last accepted policy. Thus, ϵ, K, and κ control surrogate saturation, batch reuse, and policy change, respectively. Algorithm 1 summarizes the update and Appendix D gives numerical conventions.

5 THEORETICAL PROPERTIES

Let $\mathbf{F} = \mathbf{F}_k$ be the positive-definite damped Fisher matrix in Eq. (2), with induced norm $\|\mathbf{x}\|_{\mathbf{F}}^2 = \langle \mathbf{x}, \mathbf{F}\mathbf{x} \rangle$. The residual $\mathbf{r} = \mathbf{g}_{\mathrm{s}}^{(k)} - \mathbf{F}\mathbf{d}_{\mathrm{s}}$ measures the error in the conjugate-gradient solve and vanishes for an exact solve. The following proposition characterizes the projection and its effect on the regularized surrogate. All proofs are provided in Appendix B.

Proposition 1 (Damped-Fisher projection and first-order compatibility)**.** *Let $\mathbf{d}_{\mathrm{s}} \neq 0$ and let $\bar{\mathbf{d}}_{\mathrm{v}}$ be defined by Eq. (7). Then (i) $\bar{\mathbf{d}}_{\mathrm{v}}$ is the unique minimizer of $\|\mathbf{x} - \mathbf{d}_{\mathrm{v}}\|_{\mathbf{F}}$ over the half-space $\{\mathbf{x} : \langle \mathbf{d}_{\mathrm{s}}, \mathbf{F}\mathbf{x} \rangle \geq 0\}$; (ii) $\langle \mathbf{d}_{\mathrm{s}}, \mathbf{F}\bar{\mathbf{d}}_{\mathrm{v}} \rangle = \max\{0, \chi\} \geq 0$, where $\chi = \langle \mathbf{d}_{\mathrm{s}}, \mathbf{F}\mathbf{d}_{\mathrm{v}} \rangle$; and (iii) for $\mathbf{d} = \mathbf{d}_{\mathrm{s}} + \beta\bar{\mathbf{d}}_{\mathrm{v}}$ with any $\beta \geq 0$ and $\Delta\boldsymbol{\theta} = c_\kappa \mathbf{d}$ with any $c_\kappa > 0$,*

$$\left\langle g_s^{(k)}, \Delta\boldsymbol{\theta} \right\rangle = c_\kappa \Big(\|\mathbf{d}_{\mathrm{s}}\|_{\mathbf{F}}^2 + \beta \max(0, \chi) + \langle \mathbf{r}, \mathbf{d} \rangle \Big) \geq c_\kappa \Big(\|\mathbf{d}_{\mathrm{s}}\|_{\mathbf{F}}^2 - \|\mathbf{r}\|_2 \, \|\mathbf{d}\|_2 \Big). \tag{9}$$

With an exact solve ($\mathbf{r} = \mathbf{0}$), the lower bound is $c_\kappa \|\mathbf{d}_{\mathrm{s}}\|_{\mathbf{F}}^2 > 0$. Thus, the refined direction strictly increases the regularized surrogate to first order, regardless of the nonnegative refinement weight. For an inexact solve, ascent holds whenever $\langle \mathbf{r}, \mathbf{d} \rangle > -\|\mathbf{d}_{\mathrm{s}}\|_{\mathbf{F}}^2 - \beta \max\{0, \chi\}$.

Equation (9) characterizes first-order ascent and its sensitivity to solve error. For finite updates, Eq. (8) enforces nondecrease of the batch surrogate and both empirical KL constraints.

Calibrated trust. The trust weight reflects how consistently a critic direction agrees with the paired return probes. For a fixed calibration bin b, let $p_b = \Pr(\Delta G_{\mathrm{CRN}} > 0 \mid b)$. When zero labels have probability zero, the margin $2p_b - 1$ equals the expected sign of the probe label. If label sign and magnitude are also independent within the bin, Proposition 3 gives $\mathbb{E}[\Delta G_{\mathrm{CRN}} \mid b] = (2p_b - 1)\mathbb{E}[|\Delta G_{\mathrm{CRN}}| \mid b]$, linking this margin to the expected directional probe response. Independent calibration labels yield a Hoeffding bound on the estimated margin. These results motivate the probability-margin weighting in Eq. (4). The concentration bound applies to fixed bins; VPPO uses an empirical estimate with evolving bins and the positive floor $q_{\min}$.

Rollout-level bounds. The rollout-relative KL constraint controls policy change across successive updates on the same batch. Following Kakade & Langford (2002); Schulman et al. (2015); Achiam et al. (2017), Proposition 2 relates this constraint to expected return at the population level. Under bounded rewards and a population KL budget, the performance lower bound consists of a clipped-surrogate term minus a penalty proportional to the square root of that budget. The bound establishes improvement when the surrogate term exceeds the penalty. VPPO enforces the empirical KL constraint with budget $(1 + \tau)K\kappa$; transferring the population result to a finite batch additionally requires control of estimation error.

Lemma 2 complements this bound: the cumulative refinement-target norm at each state is at most δ, independently of K. The target cap controls refinement magnitude, while the KL constraints control policy displacement.

6 EXPERIMENTAL SETUP

We evaluate VPPO against on-policy and off-policy methods on simulated control tasks and use ablations to examine the contributions of clipping and pathwise refinement.

Tasks. The experiments cover two task families implemented in mjlab (Zakka et al., 2026), a GPU-parallel framework based on MuJoCo (Todorov et al., 2012) with manager-based task definitions (Mittal et al., 2023). The first consists of Unitree G1 humanoid velocity tracking on flat and rough terrain, using the parallel locomotion setup of Rudin et al. (2022). On rough terrain, a distance-based curriculum advances the policy to more difficult terrain levels. The second family comprises six DeepMind-Control-style tasks (Tassa et al., 2018): walker run, walk, and stand; cartpole balance and swingup; and cheetah run. These tasks are ported to the same simulation framework and retain their raw reward scales.

On-policy baselines. We compare VPPO with the RSL-RL implementation of PPO (Schwarke et al., 2025) and our reproduction of REPPO (Voelcker et al., 2026). PPO retains its reference clip range, epoch and minibatch counts, and KL-adaptive learning rate. PPO and VPPO share the network architecture, advantage estimator, batch size, and environment budget. REPPO uses its own action-conditioned critic and entropy-augmented targets with its reference hyperparameters, at the same environment budget. The refinement ablation sets $\beta = 0$ and omits probes, retaining the re-linearized clipped natural-gradient update (Section 7.2).

Training configuration. The on-policy methods use 4096 environments for 5000 iterations on G1 and 2048 environments for 1500 iterations on the DMC-style tasks. VPPO uses up to $K = 5$ inner updates per rollout, matching PPO's epoch count, with KL budget $\kappa = 0.01$, clip width $\epsilon = 0.2$, and probes every ten iterations. In the main comparisons, PPO uses entropy coefficient $\alpha_H = 0.01$ on G1 and 0.005 on the DMC-style tasks; VPPO uses $\alpha_H = 0.005$ on both task families. An additional G1 comparison uses a shared entropy coefficient to assess the effect of entropy regularization (Section 7.2). Full policy, value, entropy, and KL settings are provided in Appendix D and the released configurations.

Off-policy reference methods. We also evaluate local adaptations of FastTD3 and FastSAC (Seo et al., 2025), with FastSAC based on SAC (Haarnoja et al., 2018a). On the DMC-style tasks, these methods use 128 environments, a replay buffer of 10^6 transitions, and 3000 iterations; the G1 flat-terrain runs use 256 environments and three seeds. These comparisons provide context for sample and computational efficiency under different parallelism and data-reuse regimes. Appendix G gives their configurations.

Evaluation metrics. The DMC-style comparisons use five seeds per method; seed counts for the G1 on-policy comparisons are listed in Table 1. Final performance is measured by the mean training reward over the last 100 iterations, and iterations to a threshold are computed from a 20-iteration running mean. For rough terrain, attained terrain level is the primary metric and is reported alongside reward, since the curriculum changes task difficulty during training. The final VPPO flat- and rough-terrain policies and the PPO rough-terrain policy are additionally evaluated using deterministic mean actions on a fixed command grid (Appendix H).

7 Results

7.1 Clipping under explicit KL constraints

We evaluate the role of clipping through a matched-budget ablation on walker-run, varying one component of the update at a time (Figure 4(b); Table 5, Appendix F). Taking $K = 5$ re-linearized steps on the same batch without clipping yields a final reward of 661. The rollout-relative KL remains between 0.024 and 0.028 nats, below the cap of 0.0525, while the policy standard deviation increases, reaching $\sigma \in [0.44, 0.58]$, compared with $[0.24, 0.31]$ for PPO. This pattern associates the performance loss with unsaturated per-sample ratios despite the bounded KL divergence. Restoring clipping within the same trust region recovers PPO-level performance (854 ± 19 versus 845 ± 43), with less KL displacement per rollout (0.005 nats) than a single natural step (0.006 to 0.008). Repeated evaluation of the clipped gradient thus closes the performance gap while limiting policy displacement. With probe-calibrated refinement, VPPO achieves 830 ± 29 over five seeds, comparable to PPO on this low-trust task. Using a refinement cap that scales with K yields 812, which is 42 points below the clipped updater without refinement (854), motivating the K-invariant target budget in Lemma 2.

7.2 Contribution of probe-calibrated refinement

Probe-calibrated refinement provides the largest gains on tasks with high calibrated trust. On walker-run, held-out probes indicate near-chance directional agreement, with small trust weights late in training ($q \approx 0.08$ to 0.23). VPPO achieves 830 ± 29 over five seeds, compared with 854 ± 19 over two seeds for the same updater without refinement. On flat terrain at $\alpha_H = 0.01$, where q reaches 0.92 to 0.94, refinement adds 0.7 points over three seeds (70.0 ± 0.5 versus 69.2 ± 0.4). On rough terrain at 0.01, refinement increases the final return from 7.7 ± 4.6 to 14.1 ± 2.5. At 0.005, the two VPPO seeds that progress through the curriculum achieve 41.3 ± 2.0, compared with 39.7 ± 5.1

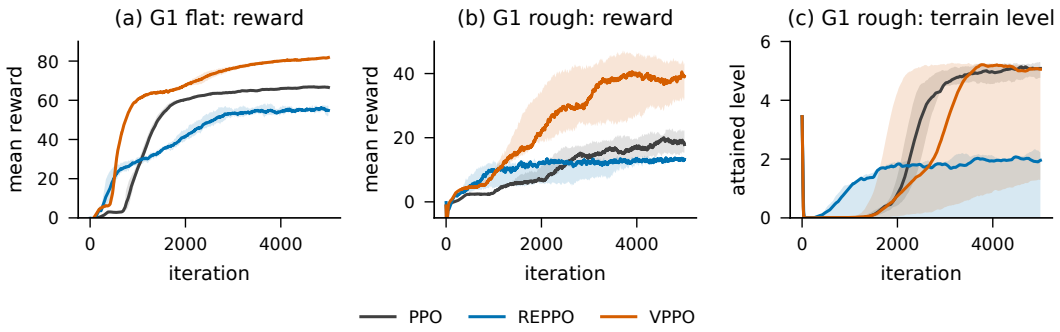


Figure 3: Unitree G1 velocity tracking against PPO (standard configuration) and REPPO; 20-iteration running means, bands are seed ranges. **(a)** flat terrain. **(b, c)** rough terrain: reward and attained terrain level (median over seeds). Two of three VPPO seeds complete the curriculum (level 5.1); the third stays on the easiest rows (Section 7.3).

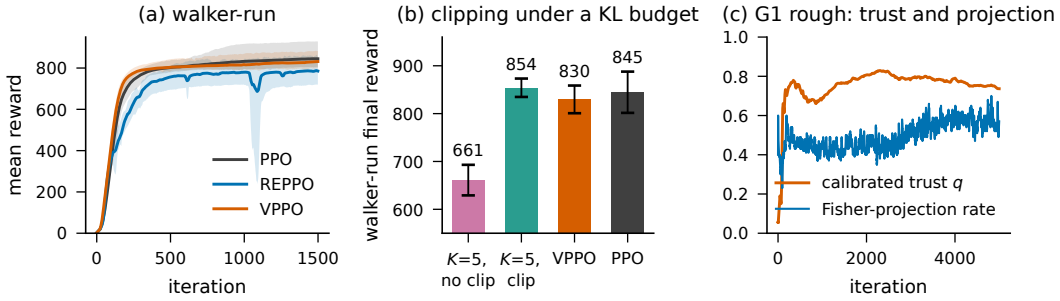


Figure 4: **(a)** Walker-run training curves (five seeds per method). **(b)** Clipping under an explicit KL budget on walker-run: final reward (mean $\pm$ standard deviation over seeds) of $K = 5$ re-linearized steps without and with the clip, VPPO, and PPO. **(c)** G1 rough terrain, seed 0: the calibrated trust rises as held-out probe agreement increases, and the damped-Fisher projection of Eq. (7) is activated on about half of the steps.

without refinement, with a lower observed standard deviation. These VPPO seeds complete the curriculum at levels 5.07 and 5.12, compared with 4.83 and 4.82 without refinement.

Figure 4(c) illustrates the evolution of calibrated trust on G1 rough terrain: the weight rises to approximately 0.8 and settles at 0.74. The damped-Fisher projection is active on 53 to 56% of rough-terrain updates, compared with 40 to 42% on flat terrain and 47 to 48% on walker-run. All G1 updates are accepted after line search, with per-step KL between 0.006 and 0.010 nats under the budget of 0.01, and rollout-relative KL between 0.014 and 0.016 under the cap of 0.0525. Backtracking reduces most step sizes; full rejection occurs only at ten times the nominal budget in the additional walker-run experiments (Appendix F).

Matched entropy coefficient. With entropy coefficients matched, VPPO reaches reward thresholds and curriculum milestones earlier than PPO. Table 6 and Figure 5 in Appendix F compare VPPO with PPO trained at VPPO's coefficient, using three PPO seeds per terrain. On flat terrain, both methods reach a reward of 81.7, but VPPO reaches intermediate reward thresholds earlier. On rough terrain, two seeds from each method make substantial curriculum progress, with the VPPO runs progressing approximately 1000 iterations earlier. Only the VPPO seeds complete the curriculum within the training budget.

7.3 Benchmark Performance and Computational Cost

Table 1 and Figures 3 and 4 summarize benchmark performance. On G1 flat terrain, VPPO achieves a final reward of 81.7, compared with 66.7 for PPO and 55.0 for REPPO. VPPO reaches reward 40 after 607 and 650 iterations, compared with 1176 to 1258 for PPO. At reward 62, the corresponding counts are 1071 and 1131 for VPPO and 2160 to 2353 for PPO. On rough terrain, VPPO achieves 38.2 ± 4.7 reward, compared with 18.7 ± 2.8 for PPO and 13.2 for REPPO. Two of three VPPO seeds complete the curriculum at level 5.1, reaching the completion threshold after 2304 and 3587

Table 1: Final performance and sample efficiency (mean $\pm$ standard deviation over seeds; seed counts for walker-run, G1 flat and G1 rough: VPPO $5, 2, 3$; PPO $5, 3, 3$; REPPO $5, 3, 3$). Rough reward is earned at the attained terrain level, listed per seed. Iterations to a threshold are per seed on a 20-iteration running mean; >5000 marks thresholds not reached within the budget. PPO is its standard configuration; the control at VPPO's entropy coefficient is in Appendix F.

	Final				G1 flat: iterations to reward		G1 rough: iterations to	
Method	Walker-run	G1 flat	G1 rough reward	G1 rough level (per seed)	40	62	level 5	reward 18
VPPO	830 ± 29	$\mathbf{81.7 \pm 0.1}$	$\mathbf{38.2 \pm 4.7}$	5.1, 5.1, 1.3	**607 / 650**	**1071 / 1131**	**2304 / 3587** / >5000	**1313 / 1818** / 1862
PPO	$\mathbf{845 \pm 43}$	66.7 ± 0.4	18.7 ± 2.8	5.1, 5.3, 4.8	1176–1258	2160–2353	2913 / 3830 / >5000	2937 / 4365 / >5000
REPPO	785 ± 54	55.0 ± 2.0	13.2 ± 0.2	0.0, 1.9, 2.3	1760–2027	>5000	>5000	>5000

Table 2: DMC-style suite: final return (mean $\pm$ standard deviation over five seeds; boldface marks the best mean). The last two columns compare the faster off-policy method with PPO on the median environment steps and wall-clock time to reach 90% of PPO's final return (Appendix G).

	On-policy, 2048 environments			Off-policy, 128 environments		Off-policy vs. PPO	
Task	PPO	REPPO	VPPO	FastTD3	FastSAC	fewer env. steps	more wall-clock
walker-run	$\mathbf{845 \pm 43}$	785 ± 54	830 ± 29	831 ± 59	784 ± 81	6×	6×
walker-walk	993 ± 1	989 ± 4	994 ± 1	$\mathbf{994 \pm 1}$	993 ± 2	12×	3×
walker-stand	$\mathbf{999 \pm 1}$	993 ± 6	998 ± 1	$\mathbf{999 \pm 1}$	$\mathbf{999 \pm 1}$	15×	3×
cheetah-run	698 ± 75	717 ± 18	724 ± 10	799 ± 54	$\mathbf{818 \pm 22}$	11×	10×
cartpole-swingup	$\mathbf{972 \pm 3}$	689 ± 314	968 ± 7	961	961 ± 11	13×	5×
cartpole-balance	**1000**	991	**1000**	985	992	12×	5×

iterations. They finish with rewards of 39.3 and 43.3 and 1.09 and 0.67 falls per episode, respectively; their mean reward is 2.2 times PPO's mean. The third VPPO seed remains at level 1.3. PPO reaches level 5 in two of three seeds after 2913 and 3830 iterations, while no REPPO seed reaches this level. In deterministic evaluation on the fixed command grid, the final VPPO flat- and rough-terrain policies record zero falls, whereas the evaluated PPO rough-terrain policy records three falls (Appendix H).

DMC-style suite and off-policy comparisons. Table 2 reports the six-task suite over five seeds per method, including FastTD3 (Seo et al., 2025) and FastSAC under their off-policy configurations (Appendix G, Figures 9 and 10). Among the on-policy methods, VPPO achieves final returns comparable to PPO across all tasks and is the only method with no observed training collapse. REPPO exhibits persistent collapse on cartpole-swingup in two of five seeds, while PPO exhibits transient collapse on cartpole-balance. The off-policy methods reach 90% of PPO's final return using 6 to 15 times fewer environment steps and outperform all on-policy methods on cheetah-run. With 128 environments, however, they require 3 to 10 times more wall-clock time than PPO with 2048 environments. VPPO reaches the thresholds sooner than the off-policy methods on five of the six tasks. The on-policy efficiency comparisons use matched parallelism.

8 Discussion, Limitations, and Conclusion

VPPO combines explicit control of policy change with probe-calibrated critic refinement. Our experiments supported complementary roles for clipping and KL constraints during repeated batch reuse. VPPO improved over standard PPO and REPPO on G1, completed the rough-terrain curriculum in two of three seeds, and achieved PPO-level returns across six DMC-style tasks without observed training collapse. At a matched entropy coefficient, its main advantage was faster learning.

Limitations. VPPO relies on counterfactual probes from cloneable simulator states that share randomness, which limits its direct application to physical systems. Its verification signal is itself approximate because probe returns are finite-horizon and critic-bootstrapped, while calibration is learned from an evolving empirical distribution. The theoretical results establish first-order compatibility and population-level KL guarantees, but finite-batch acceptance with approximate Fisher-solve does not imply monotonic improvement in the true return. Finally, the G1 study uses few seeds, and a single rough-terrain run fails to complete the curriculum, so robustness across embodiments and real-world disturbances remains open. VPPO also incurs additional computation from probes, critics, and Fisher solves.

REPRODUCIBILITY STATEMENT

The VPPO update is specified in Eqs. (1)–(8) and summarized in Algorithm 1. Appendix B provides the complete theoretical statements and proofs. Appendix D documents the hyperparameters, baseline configurations, probe collection and calibration procedures, and computational costs. The entropy controller is described in Appendix E. Section 6 specifies the evaluation protocol, with further details on deterministic policy evaluation in Appendix H.

The supplementary materials include the anonymized code archive `vppo_code_iclr.zip`, which builds on mjlab and contains the VPPO implementation, baseline implementations, task definitions, configuration files, and training and evaluation scripts. The `README.md` documents software requirements, installation, and commands for reproducing the main comparisons, ablations, and deterministic policy evaluations.

AI DISCLOSURE

In this work, we used generative AI tools (e.g., Claude, ChatGPT) to revise passages and correct grammar, as well as to search for references (only at the beginning when establishing the ideas). All research ideas, methods, experiments, analyses, and conclusions are the authors' own, and the authors reviewed and take full responsibility for all content of this paper.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations (ICLR)*, 2018.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31. PMLR, 2017.

Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998. doi: 10.1162/089976698300017746.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. What matters for on-policy deep actor-critic methods? A large-scale study. In *International Conference on Learning Representations (ICLR)*, 2021.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. doi: 10.1017/CBO9780511804441.

Karl W. Cobbe, Jacob Hilton, Oleg Klimov, and John Schulman. Phasic policy gradient. In *International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2020–2027. PMLR, 2021.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley, 2nd edition, 2006. doi: 10.1002/047174882X.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep RL: A case study on PPO and TRPO. In *International Conference on Learning Representations (ICLR)*, 2020.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 2018.

Paul Glasserman. *Monte Carlo methods in financial engineering*. Springer, 2004. doi: 10.1007/978-0-387-21617-1.

Shixiang Gu, Timothy Lillicrap, Zoubin Ghahramani, Richard E. Turner, and Sergey Levine. Q-Prop: Sample-efficient policy gradient with an off-policy critic. In *International Conference on Learning Representations (ICLR)*, 2017.

Swaminathan Gurumurthy, Zachary Manchester, and J. Zico Kolter. Practical critic gradient based actor critic for on-policy reinforcement learning. In *Learning for Dynamics and Control Conference (L4DC)*, volume 211 of *Proceedings of Machine Learning Research*, pp. 625–638. PMLR, 2023.

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. PMLR, 2018a.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018b.

Nicolas Heess, Greg Wayne, David Silver, Timothy Lillicrap, Yuval Tassa, and Tom Erez. Learning continuous control policies by stochastic value gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015.

Chloe Ching-Yun Hsu, Celestine Mendler-Dünner, and Moritz Hardt. Revisiting design choices in proximal policy optimization. *arXiv preprint arXiv:2009.10897*, 2020.

Shengyi Huang, Rousslan Fernand Julien Dossa, Antonin Raffin, Anssi Kanervisto, and Weixun Wang. The 37 implementation details of proximal policy optimization. *ICLR Blog Track*, 2022. URL https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/.

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

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 267–274, 2002.

Sham M. Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 14, 2001.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2016.

Hao Liu, Yihao Feng, Yi Mao, Dengyong Zhou, Jian Peng, and Qiang Liu. Action-dependent control variates for policy optimization via Stein's identity. In *International Conference on Learning Representations (ICLR)*, 2018.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac Gym: High performance GPU-based physics simulation for robot learning. In *NeurIPS Datasets and Benchmarks Track*, 2021.

James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2408–2417. PMLR, 2015.

Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1928–1937. PMLR, 2016.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte Carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132):1–62, 2020.

Andrew Y. Ng and Michael I. Jordan. PEGASUS: A policy search method for large MDPs and POMDPs. In *Uncertainty in Artificial Intelligence (UAI)*, pp. 406–415, 2000.

Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer, 2nd edition, 2006. doi: 10.1007/978-0-387-40065-5.

Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71(7-9):1180–1190, 2008. doi: 10.1016/j.neucom.2007.11.026.

Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *AAAI Conference on Artificial Intelligence*, volume 24, pp. 1607–1612, 2010.

James Queeney, Ioannis Ch. Paschalidis, and Christos G. Cassandras. Generalized proximal policy optimization with sample reuse. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 11909–11919, 2021.

Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning (CoRL)*, volume 164 of *Proceedings of Machine Learning Research*, pp. 91–100. PMLR, 2022.

John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, and Pieter Abbeel. Trust region policy optimization. In *International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897. PMLR, 2015.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Clemens Schwarke, Mayank Mittal, Nikita Rudin, David Hoeller, and Marco Hutter. RSL-RL: A learning library for robotics research. *arXiv preprint arXiv:2509.10771*, 2025.

Younggyo Seo, Carmelo Sferrazza, Haoran Geng, Michal Nauman, Zhao-Heng Yin, and Pieter Abbeel. FastTD3: Simple, fast, and capable reinforcement learning for humanoid control. *arXiv preprint arXiv:2505.22642*, 2025.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International Conference on Machine Learning (ICML)*, volume 32 of *Proceedings of Machine Learning Research*, pp. 387–395. PMLR, 2014.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 9460–9471, 2022.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT Press, 2nd edition, 2018. URL https://mitpress.mit.edu/9780262039246/reinforcement-learning/.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

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

George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard Turner, Zoubin Ghahramani, and Sergey Levine. The mirage of action-dependent baselines in reinforcement learning. In *International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5015–5024. PMLR, 2018.

Claas A. Voelcker, Axel Brunnbauer, Marcel Hussing, Michal Nauman, Pieter Abbeel, Eric Eaton, Radu Grosu, Amir-massoud Farahmand, and Igor Gilitschenski. Relative entropy pathwise policy optimization. In *International Conference on Learning Representations (ICLR)*, pp. 35761–35790, 2026.

Yuhui Wang, Hao He, and Xiaoyang Tan. Truly proximal policy optimization. In *Uncertainty in Artificial Intelligence (UAI)*, volume 115 of *Proceedings of Machine Learning Research*, pp. 113–122. PMLR, 2020.

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

Yuhuai Wu, Elman Mansimov, Shun Liao, Roger Grosse, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Kevin Zakka, Qiayuan Liao, Brent Yi, Louis Le Lay, Koushil Sreenath, and Pieter Abbeel. mjlab: A lightweight framework for GPU-accelerated robot learning. *arXiv preprint arXiv:2601.22074*, 2026.

A Extended related work

Trust regions, natural gradients and PPO. The natural policy gradient (Kakade, 2001) applies Amari's natural gradient (Amari, 1998) to policy search and was developed into natural actor-critic methods (Peters & Schaal, 2008). Conservative policy iteration (Kakade & Langford, 2002) introduced the performance-difference argument that underlies monotone-improvement bounds; TRPO (Schulman et al., 2015) turned it into a practical KL-constrained update solved by conjugate gradients and a line search, and Achiam et al. (2017) tightened the bound to the form used in Proposition 2. Relative entropy policy search (Peters et al., 2010) and maximum a posteriori policy optimisation (Abdolmaleki et al., 2018) use alternative formulations of KL-divergence constraints to control policy updates. PPO (Schulman et al., 2017) was introduced with two variants: one based on a clipped surrogate objective and the other on an adaptive KL penalty. Combined with Adam (Kingma & Ba, 2015) and generalized advantage estimation (Schulman et al., 2016), it has become a widely used on-policy algorithm, particularly for legged locomotion in massively parallel training settings (Rudin et al., 2022; Makoviychuk et al., 2021; Schwarke et al., 2025).Studies of PPO have shown that its clipping mechanism does not guarantee satisfaction of a trust-region constraint (Ilyas et al., 2020; Wang et al., 2020), that implementation details substantially affect its performance (Engstrom et al., 2020; Huang et al., 2022), and that its design choices interact strongly (Andrychowicz et al., 2021; Hsu et al., 2020). Other work has shown that generalizing the clipping mechanism enables the reuse of samples from previous batches (Queeney et al., 2021). Scalable approximations of the Fisher matrix (Martens & Grosse, 2015; Wu et al., 2017) make natural-gradient updates computationally tractable for large neural networks. Our work contributes to this literature by experimentally isolating the two roles of PPO clipping and introducing a method that retains only the role for which the alternatives evaluated in our experiments did not provide an effective substitute.

Pathwise gradients and control variates. Mohamed et al. (2020) review score-function and pathwise gradient estimators. PPO uses a score-function estimator (Williams, 1992) to estimate policy gradients, typically with an entropy bonus to encourage exploration (Mnih et al., 2016). Pathwise gradients computed by differentiating a critic with respect to its action input underpin deterministic actor-critic methods (Silver et al., 2014; Lillicrap et al., 2016; Fujimoto et al., 2018), stochastic value gradients (Heess et al., 2015), and maximum-entropy actor-critic methods (Haarnoja et al., 2018a). Action-dependent control variates (Gu et al., 2017; Liu et al., 2018) incorporate critic gradients into on-policy policy-gradient estimates while preserving unbiasedness. However, Tucker et al. (2018) found limited additional variance reduction on the benchmarks they evaluated and attributed previously reported performance gains to implementation differences that introduced bias. REPPO (Voelcker et al., 2026) trains an action-conditioned critic on on-policy data and updates the policy using gradients through this critic, while controlling policy entropy and KL divergence. We use REPPO as our pathwise baseline. Common random numbers are a classical variance-reduction technique for comparing simulated systems (Glasserman, 2004), and Ng & Jordan (2000) used fixed random seeds for policy search.

Entropy control and benchmark integrity. Automatic temperature tuning to maintain a target entropy (Haarnoja et al., 2018b) is widely used in maximum-entropy off-policy methods and is also used in on-policy learning by REPPO. We explain how a fixed entropy coefficient determines the entropy operating point under natural-gradient updates (Lemma 3) and adopt the same form of dual controller for entropy tuning. Amodei et al. (2016) discuss reward misspecification, while Skalse et al. (2022) formalize reward gaming in terms of optimizing a proxy reward at the expense of the intended reward. The policy remains at an easy curriculum level while obtaining a high reward (Section 7.2). This resembles reward gaming, although we have not established that it meets the formal definition. In our setting, rewards are not directly comparable across curriculum levels. We therefore recommend reporting the attained curriculum level alongside reward.

B Proofs

B.1 Proof of Lemma 1

The bound follows by bounding each clipped surrogate term using the upper clipping boundary for positive advantages and the lower clipping boundary for negative advantages, then applying the

zero-mean property of the centered advantages. Unboundedness follows by constructing a sequence of Gaussian policies whose density diverges at one sampled action with positive advantage and vanishes at all other sampled state-action pairs.

Proof. Fix a sample i and denote $\rho = \rho_{\boldsymbol{\theta}}(\mathbf{s}_i, \mathbf{a}_i)$ and $\bar{\rho} = \mathrm{clip}(\rho, 1-\epsilon, 1+\epsilon)$. Since $1-\epsilon \le \bar{\rho} \le 1+\epsilon$, the clipped surrogate term satisfies

$$\min(\rho\hat{A}_i, \bar{\rho}\hat{A}_i) \le \begin{cases} (1+\epsilon)\hat{A}_i, & \hat{A}_i > 0, \\ (1-\epsilon)\hat{A}_i, & \hat{A}_i < 0, \\ 0, & \hat{A}_i = 0. \end{cases}$$

For positive advantages, this follows from $\min(\rho, \bar{\rho}) \le 1+\epsilon$. For negative advantages, it follows from $\max(\rho, \bar{\rho}) \ge 1-\epsilon$, with the inequality reversed when multiplying by $\hat{A}_i < 0$. Thus, each term is bounded above by $\hat{A}_i + \epsilon|\hat{A}_i|$. Summing over the batch and using $\sum_i \hat{A}_i = 0$ gives

$$L^{\mathrm{clip}}(\boldsymbol{\theta}) \le \frac{1}{B}\sum_i \big(\hat{A}_i + \epsilon|\hat{A}_i|\big) = \frac{\epsilon}{B}\sum_i |\hat{A}_i|. \tag{10}$$

For the gradient claim, suppose that $\hat{A}_i > 0$ and $\rho_{\boldsymbol{\theta}}(\mathbf{s}_i, \mathbf{a}_i) > 1+\epsilon$. By continuity of the likelihood ratio in $\boldsymbol{\theta}$, this strict inequality holds in a neighbourhood of $\boldsymbol{\theta}$. The corresponding clipped surrogate term is therefore locally constant, with value $(1+\epsilon)\hat{A}_i$, and its gradient is zero. The same argument applies when $\hat{A}_i < 0$ and $\rho_{\boldsymbol{\theta}}(\mathbf{s}_i, \mathbf{a}_i) < 1-\epsilon$, in which case the term has the constant value $(1-\epsilon)\hat{A}_i$.

To establish unboundedness, choose a sample i with $\hat{A}_i > 0$. Assume that the mean network admits parameters satisfying $\boldsymbol{\mu}(\mathbf{s}_i) = \mathbf{a}_i$ and $\boldsymbol{\mu}(\mathbf{s}_j) \ne \mathbf{a}_j$ for all $j \ne i$. Fix these mean parameters and consider a sequence of Gaussian policies with covariance $\sigma^2 I$, where $\sigma \to 0$. Let $\boldsymbol{\theta}_\sigma$ denote the resulting policy parameters, and assume that $\pi_{\mathrm{o}}(\mathbf{a}_j \mid \mathbf{s}_j)$ is finite and strictly positive for every sample. Then

$$\begin{aligned} \rho_{\boldsymbol{\theta}_\sigma}(\mathbf{s}_i, \mathbf{a}_i) &= \frac{(2\pi\sigma^2)^{-n_A/2}}{\pi_{\mathrm{o}}(\mathbf{a}_i \mid \mathbf{s}_i)} \xrightarrow[\sigma\to 0]{} +\infty, \\ \rho_{\boldsymbol{\theta}_\sigma}(\mathbf{s}_j, \mathbf{a}_j) &= \frac{(2\pi\sigma^2)^{-n_A/2} \exp\!\Big(-\frac{\|\mathbf{a}_j - \boldsymbol{\mu}(\mathbf{s}_j)\|^2}{2\sigma^2}\Big)}{\pi_{\mathrm{o}}(\mathbf{a}_j \mid \mathbf{s}_j)} \xrightarrow[\sigma\to 0]{} 0, \qquad j \ne i. \end{aligned} \tag{11}$$

For each $j \ne i$, the squared distance in the exponent is strictly positive, so the exponential decay dominates the polynomial growth of the prefactor. Since $\hat{A}_i > 0$ and the batch is finite,

$$L^{\mathrm{IS}}(\boldsymbol{\theta}_\sigma) = \frac{1}{B}\sum_j \rho_{\boldsymbol{\theta}_\sigma}(\mathbf{s}_j, \mathbf{a}_j)\hat{A}_j \xrightarrow[\sigma\to 0]{} +\infty.$$

□

B.2 Proof of Lemma 3

The score of a Gaussian with respect to a log standard deviation is a centred chi-square variable whose variance is two and whose correlation with the score of the mean vanishes by symmetry; this makes the Fisher block of $\boldsymbol{\psi}$ constant and decoupled from the mean parameters, and the entropy is linear in $\boldsymbol{\psi}$.

Proof. For the diagonal Gaussian, $\log \pi_{\boldsymbol{\theta}}(\mathbf{a} \mid \mathbf{s}) = -\sum_{j=1}^{n_A}\Big(\psi_j + \frac{(a_j - \boldsymbol{\mu}_{\boldsymbol{\theta},j}(\mathbf{s}))^2}{2\sigma_j^2}\Big) - \frac{n_A}{2}\log 2\pi$ with $\sigma_j = e^{\psi_j}$. Writing $\boldsymbol{\xi}_j = (a_j - \boldsymbol{\mu}_{\boldsymbol{\theta},j}(\mathbf{s}))/\sigma_j$, which is standard normal and independent across j when $\mathbf{a} \sim \pi_{\boldsymbol{\theta}}(\cdot \mid \mathbf{s})$, the scores are

$$\frac{\partial \log \pi_{\boldsymbol{\theta}}}{\partial \psi_j} = \boldsymbol{\xi}_j^2 - 1, \qquad \frac{\partial \log \pi_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}_{\boldsymbol{\mu}}} = J_{\boldsymbol{\mu}}(\mathbf{s})^\top\big(\boldsymbol{\xi} \oslash \sigma\big), \tag{12}$$

where $J_{\boldsymbol{\mu}}(\mathbf{s})$ is the Jacobian of $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{s})$ with respect to the mean parameters $\boldsymbol{\theta}_{\boldsymbol{\mu}}$ and $\oslash$ is elementwise division. The Fisher information at $\boldsymbol{\theta}$ is the expectation of the outer product of the score under

$\mathbf{a} \sim \pi_{\boldsymbol{\theta}}(\cdot \mid \mathbf{s})$, averaged over states; the batch-averaged KL Hessian of Eq. (2) equals this quantity with the state average taken over the batch, by the standard identity $\nabla^2_{\boldsymbol{\theta}'} \mathrm{KL}(\pi_{\boldsymbol{\theta}} \| \pi_{\boldsymbol{\theta}'})|_{\boldsymbol{\theta}'=\boldsymbol{\theta}} = F(\boldsymbol{\theta})$, plus the damping $\lambda \mathbf{I}$. For the $\boldsymbol{\psi}$ block, $\mathbb{E}[(\boldsymbol{\xi}_j^2 - 1)(\boldsymbol{\xi}_l^2 - 1)] = \delta_{jl} \operatorname{Var}(\boldsymbol{\xi}_j^2) = 2\delta_{jl}$, since $\boldsymbol{\xi}_j^2$ is chi-square with one degree of freedom. For the cross block, $\mathbb{E}[(\boldsymbol{\xi}_j^2 - 1)\, \boldsymbol{\xi}_l] = \mathbb{E}[\boldsymbol{\xi}_j^2 \boldsymbol{\xi}_l] - \mathbb{E}[\boldsymbol{\xi}_l] = 0$ for all j, l by the symmetry of the standard normal, and the state average of a vector whose action expectation vanishes at every state is zero. Hence

$$F = \begin{pmatrix} F_{\boldsymbol{\mu}} & 0 \\ 0 & 2\mathbf{I}_{n_A} \end{pmatrix} + \lambda \mathbf{I}, \tag{13}$$

for every state distribution. The differential entropy is $H(\pi_{\boldsymbol{\theta}}) = \sum_j \left(\psi_j + \frac{1}{2}\log(2\pi e)\right)$, so $\nabla_{\boldsymbol{\psi}} H = \mathbf{1}$ and the natural gradient of $\alpha_H H$ in the $\boldsymbol{\psi}$ block is $(2+\lambda)^{-1}\alpha_H \mathbf{1}$, which is $\frac{\alpha_H}{2}\mathbf{1}$ for $\lambda = 0$; the mean block does not enter because of the block-diagonal structure. For the importance-sampled surrogate, $\nabla_{\boldsymbol{\psi}} L^{\mathrm{IS}} = \frac{1}{B}\sum_i \rho_i \hat{A}_i \nabla_{\boldsymbol{\psi}} \log \pi_{\boldsymbol{\theta}}(a_i \mid s_i) = \frac{1}{B}\sum_i \rho_i \hat{A}_i (\boldsymbol{\xi}_i^2 - \mathbf{1})$ with $\boldsymbol{\xi}_i$ standardized by the current policy, and the natural gradient is this vector divided by $2 + \lambda$. For the clipped surrogate the same expression holds with the sum restricted to the samples whose ratio has not crossed the clip boundary in the favoured direction (Lemma 1). □

B.3 Proof of Proposition 1

In the inner product induced by the Fisher matrix, the set of directions that do not oppose $\mathbf{d}_{\mathrm{s}}$ is a half-space with normal $\mathbf{d}_{\mathrm{s}}$, and Eq. (7) is exactly the nearest-point map onto it. The first-order statement is then a direct computation once the score gradient is written as $\mathbf{F}\mathbf{d}_{\mathrm{s}}$ plus the conjugate-gradient residual.

Proof. Equip $\mathbb{R}^n$ with the inner product $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{F}} = \langle \mathbf{x}, \mathbf{F}\mathbf{y} \rangle$, which is a genuine inner product because $\mathbf{F}$ is symmetric positive definite. The set $C = \{\mathbf{x} : \langle \mathbf{d}_{\mathrm{s}}, \mathbf{F}\mathbf{x} \rangle \geq 0\} = \{\mathbf{x} : \langle \mathbf{d}_{\mathrm{s}}, x \rangle_{\mathbf{F}} \geq 0\}$ is a closed half-space with normal $\mathbf{d}_{\mathrm{s}}$. The projection of a point $\mathbf{y}$ onto a closed half-space $\{\mathbf{x} : \langle \mathbf{n}, \mathbf{x} \rangle_{\mathbf{F}} \geq 0\}$ in a Hilbert space is

$$P(\mathbf{y}) = \mathbf{y} - \min\left(0, \langle \mathbf{n}, \mathbf{y} \rangle_{\mathbf{F}}\right) \frac{\mathbf{n}}{\|\mathbf{n}\|_{\mathbf{F}}^2}, \tag{14}$$

which we verify through the characterization of the projection onto a closed convex set, namely that $P(\mathbf{y}) \in C$ and $\langle \mathbf{y} - P(\mathbf{y}), \mathbf{x} - P(\mathbf{y}) \rangle_{\mathbf{F}} \leq 0$ for all $\mathbf{x} \in C$ (Boyd & Vandenberghe, 2004). If $\langle \mathbf{n}, \mathbf{y} \rangle_{\mathbf{F}} \geq 0$ then $P(y) = y \in C$ and the inequality is trivial. Otherwise $\langle n, P(y) \rangle_{\mathbf{F}} = \langle \mathbf{n}, \mathbf{y} \rangle_F - \langle \mathbf{n}, \mathbf{y} \rangle_{\mathbf{F}} = 0$, so $P(y) \in C$, and for any $x \in C$,

$$\langle \mathbf{y} - P(\mathbf{y}), \mathbf{x} - P(\mathbf{y}) \rangle_{\mathbf{F}} = \frac{\langle \mathbf{n}, \mathbf{y} \rangle_{\mathbf{F}}}{\|\mathbf{n}\|_{\mathbf{F}}^2} \langle \mathbf{n}, \mathbf{x} - P(\mathbf{y}) \rangle_{\mathbf{F}} = \frac{\langle \mathbf{n}, \mathbf{y} \rangle_{\mathbf{F}}}{\|\mathbf{n}\|_{\mathbf{F}}^2} \langle \mathbf{n}, \mathbf{x} \rangle_{\mathbf{F}} \leq 0, \tag{15}$$

because $\langle \mathbf{n}, \mathbf{y} \rangle_{\mathbf{F}} < 0$ and $\langle \mathbf{n}, \mathbf{x} \rangle_{\mathbf{F}} \geq 0$. Uniqueness of the nearest point is the projection theorem for closed convex sets. Setting $\mathbf{n} = \mathbf{d}_{\mathrm{s}}$, $\mathbf{y} = \mathbf{d}_{\mathrm{v}}$, $\langle \mathbf{n}, \mathbf{y} \rangle_{\mathbf{F}} = \langle \mathbf{d}_{\mathrm{s}}, \mathbf{F}\mathbf{d}_{\mathrm{v}} \rangle = \chi$ and $\|\mathbf{n}\|_{\mathbf{F}}^2 = \langle \mathbf{d}_{\mathrm{s}}, \mathbf{F}\mathbf{d}_{\mathrm{s}} \rangle$ recovers Eq. (7), which proves part (i). Part (ii) follows from $\left\langle \mathbf{d}_{\mathrm{s}}, \mathbf{F}\bar{\mathbf{d}}_{\mathrm{v}} \right\rangle = \chi - \min(0, \chi) \langle \mathbf{d}_{\mathrm{s}}, \mathbf{F}\mathbf{d}_{\mathrm{s}} \rangle / \langle \mathbf{d}_{\mathrm{s}}, \mathbf{F}\mathbf{d}_{\mathrm{s}} \rangle = \chi - \min(0, \chi) = \max(0, \chi)$.

For part (iii), write $\mathbf{g}_{\mathrm{s}}^{(k)} = F d_s + r$ and $\mathbf{d} = \mathbf{d}_{\mathrm{s}} + \beta \bar{\mathbf{d}}_{\mathrm{v}}$. Using the symmetry of F,

$$\begin{aligned} \left\langle g_s^{(k)}, d \right\rangle &= \langle \mathbf{F}\mathbf{d}_{\mathrm{s}}, \mathbf{d}_{\mathrm{s}} \rangle + \beta \left\langle \mathbf{F}\mathbf{d}_{\mathrm{s}}, \bar{\mathbf{d}}_{\mathrm{v}} \right\rangle + \langle \mathbf{r}, \mathbf{d} \rangle \\ &= \|\mathbf{d}_{\mathrm{s}}\|_{\mathbf{F}}^2 + \beta \left\langle \mathbf{d}_{\mathrm{s}}, \mathbf{F}\bar{\mathbf{d}}_{\mathrm{v}} \right\rangle + \langle \mathbf{r}, \mathbf{d} \rangle = \|\mathbf{d}_{\mathrm{s}}\|_{\mathbf{F}}^2 + \beta \max(0, \chi) + \langle \mathbf{r}, \mathbf{d} \rangle . \end{aligned} \tag{16}$$

Since $\beta \max(0, \chi) \geq 0$ and $\langle \mathbf{r}, \mathbf{d} \rangle \geq -\|\mathbf{r}\|_2 \|\mathbf{d}\|_2$ by the Cauchy-Schwarz inequality, Eq. (9) follows after multiplication by $c_\kappa > 0$. For an exact solve $r = 0$ and the bound is $c_\kappa \|\mathbf{d}_{\mathrm{s}}\|_{\mathbf{F}}^2$, which is strictly positive because F is positive definite and $\mathbf{d}_{\mathrm{s}} \neq 0$. □

B.4 Statement and proof of the population lower bound

Proposition 2 (Population performance lower bound)**.** *Assume rewards are bounded. Let $\pi = \pi_{\mathrm{o}}$ and let π' be any policy with $\mathbb{E}_{\mathbf{s}\sim d^\pi}\left[\mathrm{KL}\big(\pi(\cdot \mid \mathbf{s}) \,\|\, \pi'(\cdot \mid \mathbf{s})\big)\right] \leq \delta_{\mathrm{KL}}$. Write $\varepsilon^{\pi'} = \sup_s \left|\mathbb{E}_{\mathbf{a}\sim\pi'}[A^\pi(\mathbf{s}, \mathbf{a})]\right|$,*

finite by boundedness, and let $L^{\mathrm{clip}}_{\mathrm{pop}}(\pi') = \mathbb{E}_{\mathbf{s}\sim d^{\pi},\mathbf{a}\sim\pi}\big[\min\big(\rho\, A^{\pi}, \mathrm{clip}(\rho, 1-\epsilon, 1+\epsilon)\, A^{\pi}\big)\big]$ *with* $\rho = \pi'/\pi$. *Then*

$$J(\pi') - J(\pi) \;\geq\; \frac{1}{1-\gamma}\, L^{\mathrm{clip}}_{\mathrm{pop}}(\pi') \;-\; \frac{2\gamma\, \varepsilon^{\pi'}}{(1-\gamma)^2}\sqrt{\frac{\delta_{\mathrm{KL}}}{2}}. \tag{17}$$

For the policy $\pi_{\boldsymbol{\theta}_K}$ *produced by* K *accepted steps of Eq. (8), the batch analogues of both conditions hold with* $\delta_{\mathrm{KL}} = (1+\tau)K\kappa$ *(the clip width* ϵ *of Eq. (1) is a separate constant) and* $L^{\mathrm{clip}}(\boldsymbol{\theta}_K) \geq \alpha_H\big(H(\pi_{\boldsymbol{\theta}_0}) - H(\pi_{\boldsymbol{\theta}_K})\big)$*: the accepted update can trade surrogate for entropy only at the exchange rate* α_H.

The acceptance tests check the finite-sample analogues on rollout states with normalized advantages; they do not by themselves certify the population inequality, and at practical caps the penalty term can make the right side negative, in which case the bound is uninformative.

The argument chains three classical facts: the performance-difference bound of Achiam et al. (2017), which lower-bounds the return of a new policy by an importance-sampled surrogate minus a total-variation penalty; Pinsker's inequality together with Jensen's inequality, which turn the average total variation into the square root of the average KL divergence; and the pointwise inequality $L^{\mathrm{clip}} \leq L^{\mathrm{IS}}$, which replaces the importance-sampled surrogate by the clipped one. The batch statement then reads off the acceptance tests.

Proof. Let $\pi = \pi_{\mathrm{o}}$ and let π' satisfy the hypothesis. Corollary 1 of Achiam et al. (2017), applied with the value function V^{π} as the shaping function, states that

$$J(\pi') - J(\pi) \;\geq\; \frac{1}{1-\gamma}\, \mathbb{E}_{\mathbf{s}\sim d^{\pi},\,\mathbf{a}\sim\pi'}\big[A^{\pi}(\mathbf{s},\mathbf{a})\big] \;-\; \frac{2\gamma\, \varepsilon^{\pi'}}{(1-\gamma)^2}\, \mathbb{E}_{\mathbf{s}\sim d^{\pi}}\big[D_{\mathrm{TV}}\big(\pi'(\cdot\,|\,\mathbf{s}), \pi(\cdot\,|\,\mathbf{s})\big)\big], \tag{18}$$

where d^{π} is the normalized discounted state visitation of π and D_{TV} is the total variation distance. Pinsker's inequality (Cover & Thomas, 2006) gives $D_{\mathrm{TV}}(\pi'(\cdot\,|\,\mathbf{s}), \pi(\cdot\,|\,\mathbf{s})) \leq \sqrt{\mathrm{KL}(\pi(\cdot\,|\,\mathbf{s}) \| \pi'(\cdot\,|\,\mathbf{s}))/2}$, in which either order of the arguments of the KL divergence may be used because total variation is symmetric, and Jensen's inequality for the concave square root gives

$$\mathbb{E}_{\mathbf{s}\sim d^{\pi}}\big[D_{\mathrm{TV}}\big(\pi'(\cdot\,|\,\mathbf{s}), \pi(\cdot\,|\,\mathbf{s})\big)\big] \leq \sqrt{\tfrac{1}{2}\, \mathbb{E}_{\mathbf{s}\sim d^{\pi}}\big[\mathrm{KL}\big(\pi(\cdot\,|\,\mathbf{s}) \| \pi'(\cdot\,|\,\mathbf{s})\big)\big]} \leq \sqrt{\delta_{\mathrm{KL}}/2}. \tag{19}$$

Next, importance sampling gives $\mathbb{E}_{\mathbf{s}\sim d^{\pi},\mathbf{a}\sim\pi'}[A^{\pi}(\mathbf{s},\mathbf{a})] = \mathbb{E}_{\mathbf{s}\sim d^{\pi},\mathbf{a}\sim\pi}[\rho(\mathbf{s},\mathbf{a})A^{\pi}(\mathbf{s},\mathbf{a})]$ with $\rho = \pi'/\pi$, and for every $(\mathbf{s},\mathbf{a})$ the clipped term satisfies $\min(\rho A^{\pi}, \mathrm{clip}(\rho, 1-\epsilon, 1+\epsilon)A^{\pi}) \leq \rho A^{\pi}$ because the minimum of two numbers is at most the first. Taking expectations,

$$L^{\mathrm{clip}}_{\mathrm{pop}}(\pi') \leq \mathbb{E}_{\mathbf{s}\sim d^{\pi},\,\mathbf{a}\sim\pi'}\big[A^{\pi}(\mathbf{s},\mathbf{a})\big]. \tag{20}$$

Combining Eqs. (18) to (20) yields Eq. (17).

For the batch statement, the second test of Eq. (8) is applied at every accepted step, in particular at the last one, so $\overline{\mathrm{KL}}(\pi_{\mathrm{o}} \| \pi_{\boldsymbol{\theta}_K}) \leq (1+\tau)K\kappa$ with the batch average of Eq. (2) in place of the expectation over d^{π}; this is the batch analogue of the hypothesis with $\delta_{\mathrm{KL}} = (1+\tau)K\kappa$. The third test gives $L(\boldsymbol{\theta}_K) \geq L(\boldsymbol{\theta}_{K-1}) \geq \cdots \geq L(\boldsymbol{\theta}_0)$. At $\boldsymbol{\theta}_0 = \boldsymbol{\theta}_{\mathrm{o}}$ every ratio equals one, so $L^{\mathrm{clip}}(\boldsymbol{\theta}_0) = \frac{1}{B}\sum_i \min(\hat{A}_i, \hat{A}_i) = \frac{1}{B}\sum_i \hat{A}_i = 0$ by the zero-mean normalization, and therefore $L^{\mathrm{clip}}(\boldsymbol{\theta}_K) + \alpha_H H(\pi_{\boldsymbol{\theta}_K}) \geq \alpha_H H(\pi_{\boldsymbol{\theta}_0})$, which rearranges to the stated inequality. If the fallback ladder stops the iteration after $k' < K$ accepted steps, then $\boldsymbol{\theta}_K$ is to be read as $\boldsymbol{\theta}_{k'}$ and the same argument applies. □

B.5 Statement and proof of the sign-margin proposition

Proposition 3 (Sign margin and concentration of the calibrated trust)**.** *Fix a calibration bin* b*, consider the distribution of held-out probes* (s, a, u) *in* b *together with the probe noise, and assume* $\Pr[\Delta G_{\mathrm{CRN}} = 0 \,|\, b] = 0$ *with* $p_b = \Pr[\Delta G_{\mathrm{CRN}} > 0 \,|\, b]$*. Then* $\mathbb{E}[\mathrm{sign}(\Delta G_{\mathrm{CRN}}) \,|\, b] = 2p_b - 1$*. If, in addition,* $|\Delta G_{\mathrm{CRN}}|$ *is independent of* $\mathrm{sign}(\Delta G_{\mathrm{CRN}})$ *within the bin, then*

$$\mathbb{E}[\Delta G_{\mathrm{CRN}} \,|\, b] \;=\; (2p_b - 1)\, \mathbb{E}\big[|\Delta G_{\mathrm{CRN}}| \,\big|\, b\big], \tag{21}$$

so the unfloored weight $q_b = 2p_b - 1$ *has expected probe gain* $(2p_b-1)^2\, \mathbb{E}[|\Delta G_{\mathrm{CRN}}| \,|\, b] \geq 0$*, zero exactly at chance accuracy. If* $\hat{p}_b$ *is estimated from* n_b *independent held-out labels, then* $|\hat{q}_b - q_b| \leq \sqrt{2\log(2/\delta_c)/n_b}$ *with probability at least* $1 - \delta_c$.

The proposition characterizes the unfloored margin within a fixed calibration bin. The concentration result assumes independent labels, and the expected-gain result additionally assumes independence between label sign and magnitude, which is not evaluated empirically here. During training, both the bins and the probe distribution evolve, so the concentration result applies to fixed-bin estimation. The implemented floor $q_{\min}$ retains a positive weight at or below chance and does not inherit the nonnegative expected-gain property of the unfloored margin. A conservative variant uses a lower confidence bound, $q_b = \max\big(2\max(0, \hat{p}_b - r_b) - 1, 0\big)$, with $r_b = \sqrt{\log(1/\delta_c)/(2n_b)}$.

The expected sign of a variable that is positive with probability p and never zero is $p-(1-p) = 2p-1$, which proves the identity without any independence assumption. When the magnitude and the sign are independent, the mean factorizes into the mean magnitude times the expected sign. The concentration statement is Hoeffding's inequality for a Bernoulli mean.

Proof. Within bin b, write $\Delta G_{\mathrm{CRN}} = |\Delta G_{\mathrm{CRN}}|\,\mathrm{sign}(\Delta G_{\mathrm{CRN}})$, which is valid because $\Pr[\Delta G_{\mathrm{CRN}} = 0 \,|\, b] = 0$. By the independence assumption,

$$\mathbb{E}[\Delta G_{\mathrm{CRN}} \,|\, b] = \mathbb{E}\big[|\Delta G_{\mathrm{CRN}}| \,\big|\, b\big]\,\mathbb{E}\big[\,\mathrm{sign}(\Delta G_{\mathrm{CRN}}) \,\big|\, b\big] = \mathbb{E}\big[|\Delta G_{\mathrm{CRN}}| \,\big|\, b\big]\big(p_b - (1-p_b)\big), \quad (22)$$

which equals $(2p_b - 1)\,\mathbb{E}[|\Delta G_{\mathrm{CRN}}| \,|\, b]$, that is, Eq. (21). Multiplying by $q_b = 2p_b - 1$ gives the expected probe gain $(2p_b - 1)^2\mathbb{E}[|\Delta G_{\mathrm{CRN}}| \,|\, b]$, which is nonnegative and equals zero exactly when $p_b = \frac{1}{2}$. For the concentration statement, $\hat{p}_b$ is the average of n_b independent Bernoulli(p_b) labels, so Hoeffding's inequality gives $\Pr[|\hat{p}_b - p_b| \geq t] \leq 2\exp(-2n_b t^2)$. Choosing $t = \sqrt{\log(2/\delta_c)/(2n_b)}$ makes the right-hand side equal to δ_c, and on the complementary event $|\hat{q}_b - q_b| = 2|\hat{p}_b - p_b| \leq 2t = \sqrt{2\log(2/\delta_c)/n_b}$. □

B.6 Statement and proof of the target cap

Lemma 2 (K-invariant cap on cumulative refinement targets)**.** *For every state* $\mathbf{s}$ *and every* $K \geq 1$*, the cumulative refinement target over one rollout satisfies* $\big\|\sum_{k=1}^{K} \mathbf{v}_k(\mathbf{s})\big\| \leq \sum_{k=1}^{K} \|\mathbf{v}_k(\mathbf{s})\| \leq \delta$ *(a bound on the* $\boldsymbol{\sigma}_{\mathrm{o}}$*-scaled target vectors), and* $K = 1$ *recovers the single-step cap* δ *exactly. The cap concerns the target vectors* $\mathbf{v}_k$*; the realized mean displacement additionally passes through the network Jacobian, the Fisher solves, the projections, the weight* β*, the KL scaling and the line search, and is controlled by the acceptance tests, not by this lemma.*

The cap bounds every summand and the triangle inequality bounds the sum.

Proof. By the definition of cap_{ϱ} in Eq. (5), $\big\|\mathrm{cap}_{\varrho}(\mathbf{u})\big\| = \|\mathbf{u}\|\min(1, \varrho/\|\mathbf{u}\|) \leq \varrho$ for every $\mathbf{u} \neq \mathbf{0}$, and $\mathrm{cap}_{\varrho}(0) = 0$, so $\|\mathbf{v}_k(\mathbf{s})\| \leq \delta/K$ for every k and every $\mathbf{s}$. The triangle inequality then gives $\Big\|\sum_{k=1}^{K} \mathbf{v}_k(\mathbf{s})\Big\| \leq \sum_{k=1}^{K} \|\mathbf{v}_k(\mathbf{s})\| \leq K \cdot \delta/K = \delta$. For $K = 1$ the cap is δ itself. □

C Algorithm

Algorithm 1 summarizes one VPPO iteration. The critic ensemble, probe streams, and calibration table are maintained across iterations. The value network used to estimate advantages is trained by minibatch regression between rollouts and is omitted from the pseudocode. Both natural directions are computed by conjugate gradients on a fixed batch subsample, while the acceptance tests use the full batch.

D Implementation details

Shared configuration. All methods share the actor and critic architectures, generalized advantage estimator, numbers of environments and training iterations, and task-specific observation and reward definitions. PPO uses the reference implementation of Schwarke et al. (2025), with five epochs of minibatch Adam updates per rollout and a clip range of 0.2. Its learning rate is multiplied or divided by a fixed factor when the measured KL divergence from the rollout policy leaves the target band; both the factor and the band are specified in the official RSL-RL configuration. VPPO uses PPO's clip range and epoch count as ϵ and K, respectively, so both methods reuse each batch equally

Algorithm 1 One VPPO iteration

Require: rollout policy $\boldsymbol{\theta}_{\mathrm{o}}$; steps K; per-step budget κ; tolerance τ; backtracking factor ζ; refinement weight β; refinement target cap δ; trust floor $q_{\min}$

Collect a batch $\mathcal{D} = \{(\mathbf{s}_i, \mathbf{a}_i, \hat{A}_i)\}_{i=1}^{B}$ with $\pi_{\boldsymbol{\theta}_{\mathrm{o}}}$; normalize $\hat{A}$ to zero mean and unit variance
Every m iterations: run common-random-number branch probes (Eq. (3)); assign each label at random to exactly one of the two disjoint streams (training or held-out); refresh the calibration table $\hat{p}(\cdot)$ from the held-out stream
Train the critic ensemble $\{Q_j\}$ on $\mathcal{D}$ by n-step SARSA regression plus the Huber directional loss on the training stream; update the Polyak targets $\bar{Q}$
Compute the trust $q(\mathbf{s}_i, \mathbf{a}_i)$ for every sample by Eq. (4)
$\boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta}_{\mathrm{o}}$
for $k = 0, \ldots, K-1$ **do**
    $\mathbf{g}_{\mathrm{s}} \leftarrow \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}_k)$ ▷ clipped surrogate, ratios against $\pi_{\boldsymbol{\theta}_{\mathrm{o}}}$, Eq. (1)
    $\mathbf{d}_{\mathrm{s}} \leftarrow \mathrm{CG}(\mathbf{F}_k, \mathbf{g}_{\mathrm{s}})$ ▷ Fisher-vector products at $\boldsymbol{\theta}_k$, Eq. (2)
    if $\mathbf{d}_{\mathrm{s}} = 0$: **stop** consuming this batch ▷ step normalization undefined
    Compute $\mathbf{v}_k(\mathbf{s}_i)$ for every sample by Eq. (5); $\mathbf{g}_{\mathrm{v}} \leftarrow \nabla_{\boldsymbol{\theta}} R_k(\boldsymbol{\theta}_k)$ by Eq. (6), with q and $\mathbf{v}_k$ detached
    $\mathbf{d}_{\mathrm{v}} \leftarrow \mathrm{CG}(\mathbf{F}_k, \mathbf{g}_{\mathrm{v}})$; $\chi \leftarrow \langle \mathbf{d}_{\mathrm{s}}, \mathbf{F}_k \mathbf{d}_{\mathrm{v}} \rangle$; $\bar{\mathbf{d}}_{\mathrm{v}} \leftarrow \mathbf{d}_{\mathrm{v}} - \min(0, \chi)\, \mathbf{d}_{\mathrm{s}} / \langle \mathbf{d}_{\mathrm{s}}, \mathbf{F}_k \mathbf{d}_{\mathrm{s}} \rangle$ ▷ Eq. (7)
    accepted $\leftarrow$ LINESEARCH$(\boldsymbol{\theta}_k, \mathbf{d}_{\mathrm{s}} + \beta \bar{\mathbf{d}}_{\mathrm{v}})$
    if not accepted **then** accepted $\leftarrow$ LINESEARCH$(\boldsymbol{\theta}_k, \mathbf{d}_{\mathrm{s}})$ ▷ fallback to the score direction
    end if
    if not accepted **then break** ▷ no update; stop consuming the batch
    end if
    $\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \Delta\boldsymbol{\theta}_k$
end for
Update the entropy coefficient by Eq. (23) if the closed-loop variant is used

function LINESEARCH$(\boldsymbol{\theta}, d)$
    for $\alpha \in \{1, \zeta, \zeta^2, \ldots, \zeta^9\}$ **do**
        $\Delta\boldsymbol{\theta} \leftarrow \alpha \sqrt{2\kappa / \langle d, \mathbf{F}_k d \rangle}\, d$
        if $\overline{\mathrm{KL}}(\pi_{\boldsymbol{\theta}_k} \| \pi_{\boldsymbol{\theta}_k + \Delta\boldsymbol{\theta}_k}) \le (1+\tau)\kappa$ **and** $\overline{\mathrm{KL}}(\pi_{\boldsymbol{\theta}_{\mathrm{o}}} \| \pi_{\boldsymbol{\theta}_k + \Delta\boldsymbol{\theta}_k}) \le (1+\tau)K\kappa$ **and** $L(\boldsymbol{\theta}_k + \Delta\boldsymbol{\theta}_k) \ge L(\boldsymbol{\theta}_k)$ **then**
            return accepted with $\Delta\boldsymbol{\theta}_k \leftarrow \Delta\boldsymbol{\theta}$
        end if
    end for
    return not accepted
end function

often. Table 3 summarizes the VPPO-specific settings. The released repository provides the complete configurations, including network sizes, advantage estimation, and critic-ensemble and probe settings.

Table 3: VPPO settings used in all experiments. The ablation without the refinement uses the same values with $\beta = 0$ and no probes.

Symbol	Meaning	Value
K	re-linearized natural steps per rollout	5
κ	per-step KL budget (nats)	0.01
τ	acceptance tolerance on both KL tests	0.05
$K\kappa(1+\tau)$	resulting per-rollout KL cap (nats)	0.0525
ζ	backtracking factor, at most ten trials	0.6
λ	Fisher damping	0.1
ϵ	ratio clip range (as PPO)	0.2
α_H	entropy coefficient, G1 matched comparisons and DMC-style tasks	0.005
α_H	entropy coefficient, published G1 configuration	0.01
$\sigma^\star, \eta$	controller target and adaptation rate (rough-terrain experiments); coefficient clamp $[0, 0.05]$	$0.45, 5 \times 10^{-4}$
m	probe interval (iterations)	10
T_p, h	probe horizon and perturbation size	2, 0.2
β	refinement weight (fixed)	1.0
δ	per-rollout cap on the $\boldsymbol{\sigma}_{\mathrm{o}}$-scaled refinement targets (per step δ/K)	0.1 (0.02)
$q_{\min}$	trust floor	0.05
M, n	critic-ensemble size and n-step target	4, 3
	disagreement bins, per-bin minimum, calibration window	6 quantile bins, 60 labels, ring buffer of 1024 held-out labels
	held-out fraction of probe labels; CG iterations, squared-residual tolerance, FVP subsample	0.25; 15, 10^{-10}, 25% of the batch

Probes and calibration. The probe environment is a copy of the task that supports deterministic state restoration. Two branches initialized from the same simulator snapshot and provided with the same random numbers therefore differ only in their initial action perturbation. At each refresh (every $m = 10$ iterations), we resample commands for the 256 probe environments and advance them for 40 steps under the mean policy with Gaussian action noise of scale 0.15. This procedure moves the states away from the reset distribution before collecting 32 source-state snapshots. Each source state is replicated across 2×4 slots, forming four common-random-number pairs of $+h$ and $-h$ branches. A refresh yields 32 labels from $2\,T_p \times 256$ probe-environment steps. The probe action is the policy mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{s})$ at the source state. Each probe produces one directional label ΔG_{CRN} (Eq. (3)), assigned to either the training stream or the held-out calibration stream. Training labels supervise the directional derivatives of all ensemble members through the Huber loss. Held-out labels are used exclusively to estimate $\hat{p}$ within quantile bins of the ensemble disagreement U. The streams remain disjoint, so trust estimates use labels excluded from critic training. The critic ensemble is trained with 64 Adam updates per iteration, using a learning rate of 10^{-3}, a batch size of 4096, and a Polyak averaging rate of 0.01. Training labels are stored in a first-in-first-out buffer of 2048 labels. Each ensemble update includes a sampled Huber directional loss with coefficient 1. Probe collection accounts for approximately 0.1% of the environment steps in a run.

Conjugate gradients. Both natural directions are computed using conjugate gradients, with Fisher-vector products evaluated on a fixed batch subsample and damping λ as in Eq. (2). The acceptance tests in Eq. (8) use the full batch. Proposition 1 characterizes the effect of the solve residual through $\langle r,\, d\rangle$, while the third acceptance test directly enforces nondecrease of the regularized surrogate for the finite update.

Implementation conventions. We skip a probe when $\nabla_{\mathbf{a}}\bar{Q} = 0$, define Π_{safe} as the identity when $\mathbf{w} = \mathbf{0}$, and set $\mathrm{cap}_{\varrho}(0) = 0$. If $\mathbf{d}_{\mathrm{s}} = 0$, the K-loop terminates for the current rollout because step normalization is undefined. Trust weights and refinement targets are detached when differentiating R_k. We record all quantities in Eq. (8), together with the trust weights and the projection and fallback indicators.

Computational complexity. Let N denote the number of environments, T the rollout length, $B = NT$ the batch size, P the number of policy parameters, P_Q the number of parameters in one critic, c the conjugate-gradient iteration count, and s the fraction of the batch used for Fisher-vector products. Each PPO iteration requires $O(B)$ environment steps and $O(E\,B\,P)$ computation for E epochs of minibatch updates. Each re-linearized VPPO step adds two conjugate-gradient solves, each costing $O(c\,s\,B\,P)$, at most ten line-search trials, each costing $O(B\,P)$, and per-state projections costing $O(B\,n_A)$. A Fisher-vector product requires one forward and two backward passes on the subsample, while each line-search trial requires a forward pass to evaluate the two KL constraints and the surrogate. Across K steps, the policy-update cost is $O\big(K(2cs + 10)\,B\,P\big)$, compared with PPO's $O(E\,B\,P)$. The corresponding factor is at most $K(2cs + 10)/E \approx 17.5$ under the reported settings ($c = 15$, $s = 0.25$, $K = E = 5$). In practice, the cost is substantially lower because most steps accept the first trial. Critic training adds $O(M\,U\,B_Q\,P_Q)$ computation per iteration for U updates with batch size B_Q. Probes require $2\,T_p\,N_p$ additional environment steps every m iterations, approximately 0.1% of the environment budget. Additional memory stores the M critics and their target networks, a 2048-label training buffer, and a 1024-label calibration ring buffer; no replay buffer is used. The measured per-iteration costs reported below are 1.6 to 2.3 times those of PPO, within the theoretical bound.

Timing. Table 4 reports mean per-iteration collection and learning times recorded by the training loop for the runs in Table 1. VPPO's learning time is dominated by critic-ensemble regression and the two conjugate-gradient solves per step. The variant without refinement omits the critic ensemble. Measurements were collected across sessions without process isolation, so the reported ratios are approximate. Several G1 runs shared a GPU with another run, increasing collection times by up to a factor of one and a half. Learning times were unaffected and therefore provide a more direct comparison of update costs. A controlled wall-clock-to-threshold study remains future work. Timings for the published PPO and REPPO G1 reference runs, which were trained in an earlier session, are not included.

Table 4: Per-iteration wall-clock time in seconds (mean over the run and over seeds): environment collection, learning update, and total. The last column is the total relative to PPO on the same task.

Task	Method	Collection	Learning	Total	Relative to PPO
Walker-run	PPO	0.38	0.03	0.41	1.0
	REPPO	0.38	0.44	0.82	2.0
	Without refinement	0.37	0.24	0.61	1.5
	VPPO	0.39	0.54	0.93	2.3
G1 flat	PPO ($\alpha_H = 0.005$)	0.71	0.05	0.76	1.0
	VPPO ($\alpha_H = 0.005$)	0.72	0.95	1.67	2.2
G1 rough	PPO ($\alpha_H = 0.005$)	1.45	0.06	1.51	1.0
	Without refinement ($\alpha_H = 0.005$)	1.45	0.48	1.93	1.3
	VPPO ($\alpha_H = 0.005$)	1.41	1.03	2.44	1.6

Measurement protocol. Final scores are the logged mean training rewards averaged over the last 100 iterations of each run. Across seeds, we report the mean and standard deviation. The iteration count to a threshold is the first iteration at which the 20-iteration running mean reaches that threshold. Attained terrain level is the mean curriculum level across environments, averaged over the last 100 iterations. Falls per episode are computed from the logged fall-termination rate over the same window. For deterministic policy evaluation, we run the final checkpoint with mean actions on a fixed command grid and record base height, velocity and yaw tracking errors, stance fraction, air time, landing force, action rate, and falls.

E THE ENTROPY OPERATING POINT

The entropy bonus in Eq. (1) contributes a positive, data-independent term to the update of σ, whereas the surrogate contribution can have either sign. Its contribution to the natural-gradient direction admits the following closed form.

Lemma 3 (Constant entropy pull in natural coordinates)**.** *For the diagonal Gaussian policy with state-independent $\psi = \log \boldsymbol{\sigma}$, the undamped Fisher information is block-diagonal between the mean parameters and ψ, and the ψ block equals $2\mathbf{I}_{n_A}$ for every state distribution. With the damping λ of Eq. (2), the natural gradient of $\alpha_H H(\pi_{\boldsymbol{\theta}})$ with respect to ψ equals $\frac{\alpha_H}{2+\lambda}\mathbf{1}$, independent of the batch, the state distribution and the advantages, while that of the importance-sampled surrogate with respect to ψ_j equals $\frac{1}{2+\lambda} \cdot \frac{1}{B}\sum_i \rho_i \hat{A}_i \, (\xi_{ij}^2 - 1)$, with $\boldsymbol{\xi}_i$ standardized by the current policy.*

The proof is given in Appendix B.2. The lemma describes the unscaled direction. Joint KL normalization and line search subsequently rescale the accepted step, making the realized change in ψ data-dependent. The surrogate contribution to σ is determined by the batch correlation between advantages and squared noise, which is near zero when the advantages are uninformative. In contrast, the entropy contribution is constant with respect to the data. These contributions are additive in natural coordinates, whereas Adam bounds each parameter update by a constant multiple of the learning rate, independent of gradient magnitude (Kingma & Ba, 2015). The entropy coefficient therefore shifts the operating point of σ, with sensitivity governed by the update geometry. Section 7.2 illustrates both failure modes. VPPO therefore also offers a closed-loop variant, a dual controller

$$\alpha_H \leftarrow \Pi_{[0,c_{\max}]}\Big(\alpha_H + \eta\big(H^\star - \widehat{H}(\pi_{\boldsymbol{\theta}})\big)\Big), \qquad H^\star = n_A\Big(\ln\sigma^\star + \tfrac{1}{2}\big(1 + \ln 2\pi\big)\Big), \tag{23}$$

that holds the measured entropy at a target $\sigma^\star$.

At 0.005, PPO and VPPO exhibit similar σ trajectories on both terrains (flat: 0.24 versus 0.23; rough: 0.34 versus 0.33 to 0.35), indicating that the entropy coefficient determines the operating point for both optimizers. The natural-gradient update is more sensitive to this coefficient than Adam, consistent with Lemma 3. The rough-terrain experiments without refinement illustrate the effects of higher and lower coefficients (Appendix F). At 0.01, the policy's σ increases from 0.51 to 0.56, while PPO maintains 0.51, and mean reward peaks near 9.7 before declining to 7.7 ± 4.6. At 0.002, σ decreases to 0.21, and the policy stops tracking the commanded velocity. It instead accumulates reward from the angular-tracking and action-rate terms at the lowest curriculum levels, finishing at

level 0.01 with reward 43.3. A comparison based solely on reward would therefore favor a policy that makes little curriculum progress. With target $\sigma^\star = 0.45$, the closed-loop controller in Eq. (23) adapts the coefficient to 0.0072 and matches PPO's standard-configuration rough-terrain score (18.5) at level 4.6.

F ADDITIONAL RESULTS

Decomposition ablation. Table 5 summarizes the variants evaluated in the decomposition study of Section 7.1.

Table 5: Decomposition ablation on walker-run. Final reward is reported as mean ± standard deviation across seeds, with seed counts in parentheses. Each row changes one component of the update rule at the same budget.

Updater variant	Final	What it shows
$K = 1$ natural step per rollout, $\kappa = 0.01$	780 ± 9 (3)	one step under-uses the batch
$K = 1$, budget raised to $\kappa = 0.04$	786 ± 13 (2)	step size is not the gap
$K = 5$ re-linearized steps, no clip	661 ± 32 (2)	reuse without the clip inflates σ
$K = 5$ re-linearized steps, clipped (VPPO without refinement)	854 ± 19 (2)	equals PPO (854 ± 54, 3 seeds)
plus refinement, budget scaled with K	812 ± 1 (2)	larger-cap refinement on a low-trust task
plus refinement, K-invariant budget (VPPO)	860 ± 28 (2); 830 ± 29 in five fresh seeds	parity with PPO (845 ± 43, five seeds)

Matched-coefficient control. Figure 5 compares VPPO and PPO trained with the same entropy coefficient, $\alpha_H = 0.005$, on G1, using three PPO seeds per terrain. Section 7.2 and Table 6 report the corresponding results. On flat terrain, both methods reach the same reward ceiling. VPPO reaches intermediate thresholds earlier, with the difference decreasing at higher thresholds. On rough terrain, two of three seeds make substantial curriculum progress for each method. VPPO progresses about 1000 iterations earlier, and only its seeds complete the curriculum within the training budget.

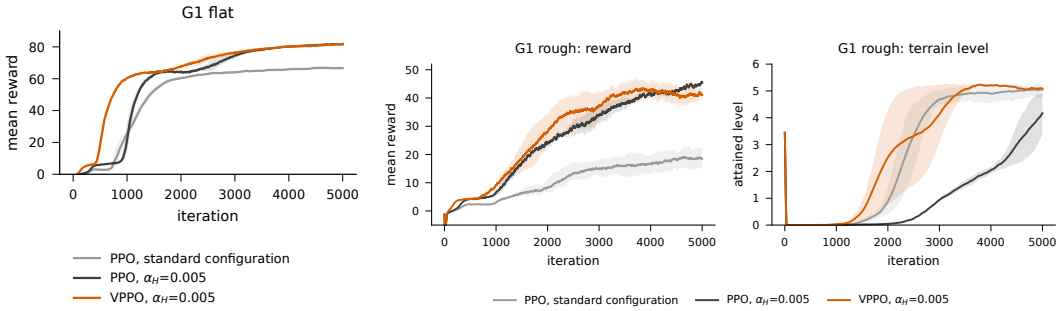


Figure 5: VPPO and PPO at the same entropy coefficient $\alpha_H = 0.005$ (20-iteration running means; bands are seed ranges; standard-configuration PPO for reference). **Left:** flat terrain. **Center and right:** rough terrain, reward and attained terrain level, averaged over the two seeds of three that engaged the curriculum for each method. Per-step reward is not comparable across terrain levels: the matched PPO seeds finish at levels 3.2 and 4.9 (reward 44.7 and 45.1) and the VPPO seeds at 5.1 (39.3 and 43.3).

Policy noise. Figure 6 shows the evolution of the policy standard deviation during training. PPO and VPPO follow similar trajectories on both terrains when the entropy coefficient is matched. Under the standard coefficient, all methods maintain a higher σ on rough terrain.

Entropy regularization on rough terrain. Figure 7 shows the effect of entropy regularization on VPPO without refinement in single-seed rough-terrain experiments. With PPO's standard entropy coefficient, σ increases while reward declines. At coefficient 0.002, σ decreases, and the policy attains a high reward while remaining at the lowest curriculum levels. In contrast, the coefficient 0.005 and the closed-loop controller both support curriculum progression. Table 7 reports the corresponding end-of-training statistics. At 0.002, the policy receives substantially less linear-tracking reward, falls more frequently, and remains at the easiest terrain levels, indicating that its higher total reward does not reflect improved locomotion.

Table 6: Matched-coefficient control on G1 ($\alpha_H = 0.005$), per seed. Flat: final reward and iterations to reward thresholds (20-iteration running mean). Rough: final attained terrain level, iteration at which the level first exceeds 0.5 (engagement), and iteration at which it reaches 5; >5000 marks events that did not occur within the budget.

		Flat				Rough		
Method	Seed	Final	to 40	to 62	to 75	Final level	Engaged at	Level 5 at
VPPO	0	81.7	650	1071	2959	5.07	1864	3587
	1	81.7	607	1131	2503	5.12	1399	2304
	2	–	–	–	–	1.29	3330	>5000
PPO	0	81.3	1095	1507	3150	3.23	2757	>5000
	1	82.2	1049	1407	2906	0.09	>5000	>5000
	2	81.5	1134	1502	3080	4.90	2696	>5000

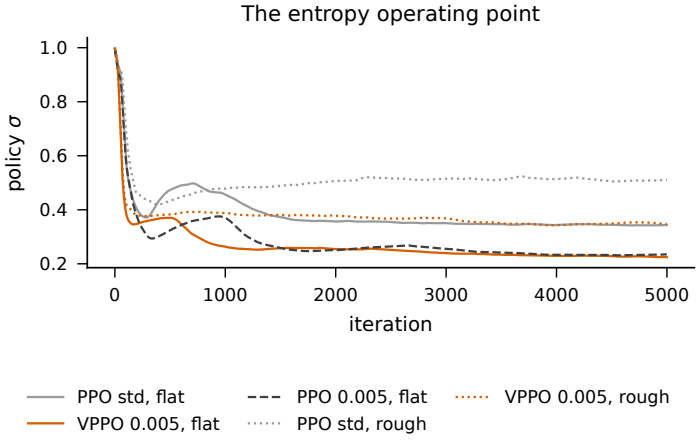


Figure 6: Policy standard deviation (mean over action dimensions) on G1 flat and rough terrain for PPO at the standard and at the matched coefficient and for VPPO at the matched coefficient. The two optimizers exhibit similar operating points when the entropy coefficient is matched.

Curriculum progression at a fixed coefficient. Table 6 reports the final terrain level of each rough-terrain seed at $\alpha_H = 0.005$. All three optimizers exhibit a bimodal pattern of curriculum progression. Two VPPO seeds complete the curriculum, while the third makes limited progress. The updater without refinement shows a similar pattern (levels 4.83, 4.82 and 1.41). Two PPO seeds begin progressing through the curriculum late in training, reaching terrain levels 3.2 and 4.9, while the third remains near the initial level. This shared pattern associates the bistability with the entropy operating point rather than a particular update rule.

KL divergence and step acceptance. Figure 8 reports the acceptance statistics for a rough-terrain VPPO run. The per-step KL divergence stays within the budget, the per-rollout divergence stays well below the cap, the accepted step scale decreases slowly over training, and neither a rejection nor a fallback occurs.

DMC-style suite. Figure 9 in Appendix G presents learning curves for all six DMC-style tasks, using five seeds per method, and Table 8 reports the corresponding numerical results. Walker-walk, walker-stand and cartpole-balance reach the same ceiling for all on-policy methods, with two transient collapses of PPO seeds on cartpole-balance (to 166 and 442, both recovered). On cartpole-swingup two REPPO seeds collapse after about 60% of training and do not recover (250 and 296 final) and a third recovers; VPPO reaches the ceiling in every seed. On cheetah-run, the three on-policy methods achieve comparable returns relative to the observed seed variability (698 ± 75, 717 ± 18 and 724 ± 10 for PPO, REPPO and VPPO; one PPO seed finishes at 554). The refinement weight remains at its floor on this low-trust task. No VPPO run exhibits a performance collapse on any task. REPPO exhibits

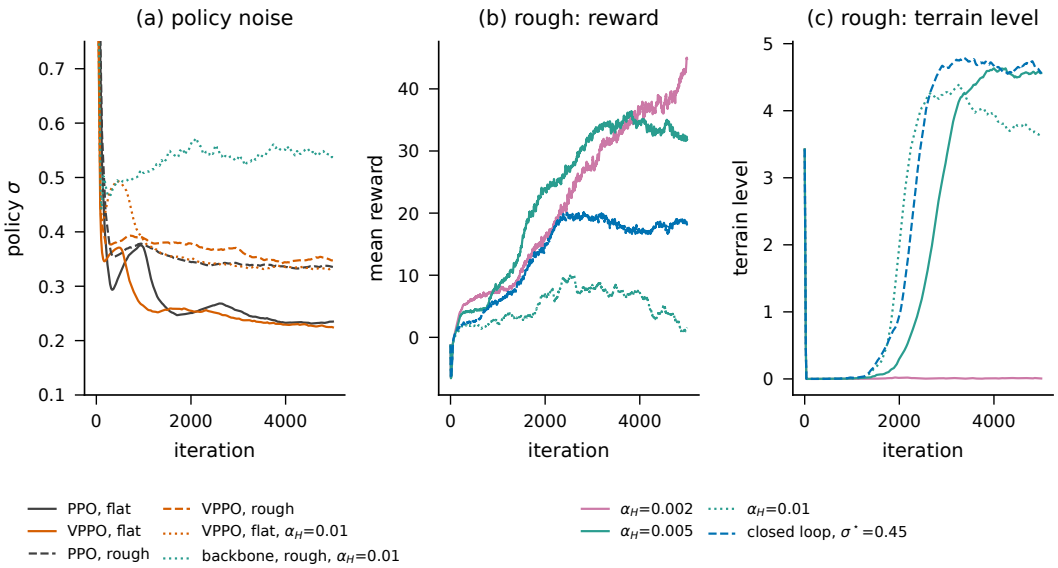


Figure 7: Effect of entropy regularization on G1 locomotion. **Left:** policy noise of PPO and VPPO at the matched coefficient on both terrains. **Center and right:** training reward and attained terrain level for VPPO without refinement on rough terrain, comparing three fixed entropy coefficients with closed-loop entropy control in single-seed experiments.

Table 7: Effect of entropy regularization on G1 rough terrain (VPPO without refinement, seed 0): end-of-training statistics for three fixed entropy coefficients and for the closed-loop controller with target $\sigma^\star = 0.45$. The last two columns are the logged per-episode reward terms averaged over the last 100 iterations: the policy at 0.002 earns less than half the linear-tracking reward of the others and a third of the action-rate penalty, consistent with a policy that remains stationary and accumulates reward at the lowest terrain levels. The closed-loop coefficient settles at 0.0072.

Entropy setting	Final reward	Terrain level	σ	Falls per episode	Linear tracking reward	Action-rate penalty
$\alpha_H = 0.01$ (PPO standard)	1.2	3.65	0.60	3.82	0.68	−1.15
$\alpha_H = 0.005$	32.0	4.57	0.36	1.09	0.95	−0.81
$\alpha_H = 0.002$	43.3	0.01	0.21	2.42	0.42	−0.29
closed loop, $\sigma^\star = 0.45$	18.5	4.60	0.47	1.40	0.94	−1.11

two additional transient collapses, on walker-run and walker-stand, alongside the two persistent collapses on cartpole-swingup.

Acceptance under larger budgets. To examine the effect of the acceptance rule at larger KL budgets, we evaluate VPPO on walker-run at four and ten times the per-step budget ($\kappa = 0.04$ and 0.10, one seed each, 1500 iterations). The backtracking line search is active at every budget: the mean accepted step scale α falls from 0.50 at $\kappa = 0.01$ (two seeds) to 0.28 at 0.04 and 0.19 at 0.10, and the realized per-step KL stays between 0.0035 and 0.0047 nats while the nominal budget grows tenfold, indicating that backtracking limits the realized step size as the nominal budget increases. Full rejection of a combined step, followed by the score-only fallback, occurs only at $\kappa = 0.10$, on 0.2% of steps in three iterations; final returns remain comparable (819 and 847 against 832 and 889). At the standard budget, the acceptance rule acts primarily through backtracking; the fallback is invoked only at the larger budget. Its causal effect under a corrupted critic has not been evaluated.

Training-window comparison at matched entropy. We additionally compare training windows within the same band of mean terrain levels at the matched entropy coefficient. This exploratory comparison retains differences in terrain distributions, training histories, and window lengths, and therefore does not constitute a controlled evaluation at matched difficulty. On rough terrain, the PPO control's reward trajectory is interpreted jointly with curriculum progression. Two control seeds begin progressing at 2696 and 2757 iterations, while the third does not progress. They reach terrain levels 3.2 and 4.9 within the 5000-iteration budget and continue progressing at its end. These trajectories

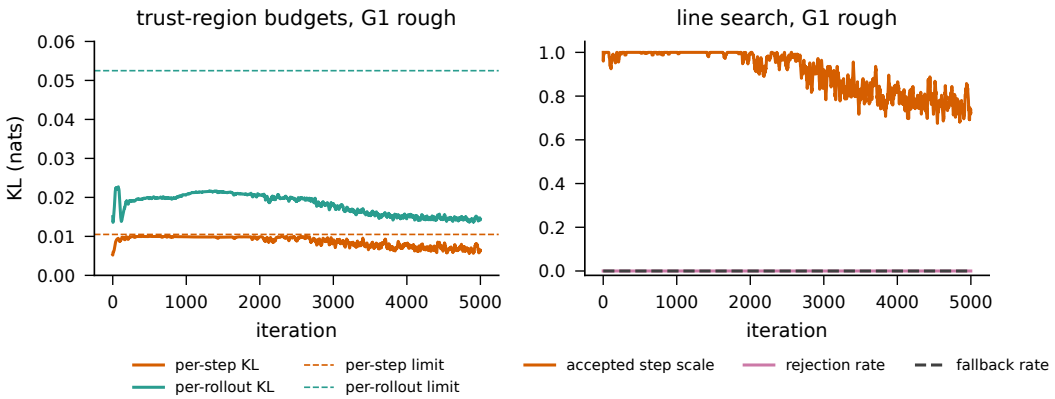


Figure 8: Acceptance statistics on G1 rough terrain (VPPO, seed 0, 20-iteration running mean). **Left:** KL divergence to the previous step and to the rollout policy against their limits. **Right:** accepted step scale α, rejection rate and fallback rate.

indicate slower learning within the available budget without establishing a performance ceiling. Every VPPO seed that progresses through the curriculum reaches 5.1 by iteration 3600. The control's raw per-step rewards (44.7 and 45.1, compared with VPPO's 39 to 43) are obtained on easier terrain for the seed at level 3.2 and at a comparable level for the seed at 4.9. For a closer comparison with the seed-0 PPO control, we consider the terrain-level band 3.0-3.4. The two progressing VPPO seeds average rewards of 31.7 ± 1.5 and 32.7 ± 1.7 (mean $\pm$ s.d. over their 53- and 75-iteration windows, ending at iterations 1840 and 3054). Their short residence times reflect rapid curriculum progression. The PPO control reaches this band only in the final $\sim$100 iterations, with reward 44.7 ± 2.0, after $\sim$2100 iterations of training at lower terrain levels since the onset of progression. These results illustrate how training history and curriculum progression affect per-step reward. A fixed-difficulty evaluation is needed for a controlled comparison of policy performance.

G Off-policy reference methods

The comparisons in Section 7 focus on on-policy methods evaluated at the same level of environment parallelism. Off-policy value-gradient methods are established baselines for environment-step efficiency, but typically operate with fewer parallel environments, a replay buffer, and multiple gradient updates per collected transition. To place our on-policy results in this broader context, we also evaluated FastTD3 and FastSAC on the six DMC-style tasks.

FastTD3 (Seo et al., 2025) builds on TD3 (Fujimoto et al., 2018), retaining twin critics, target-policy smoothing, and delayed actor updates while incorporating LayerNorm critics, HL-Gauss distributional heads, and large training batches. FastSAC builds on SAC (Haarnoja et al., 2018a), using automatic temperature tuning and the same implementation enhancements. Both implementations were validated on pendulum swing-up before evaluation. Their configurations are reported in Table 9.

We evaluated each method over five seeds on the same GPU class. The off-policy methods used 128 parallel environments, 32-step rollouts, and 3000 training iterations, corresponding to approximately 1.2×10^7 environment steps. The on-policy methods used 2048 parallel environments and 1500 training iterations, corresponding to approximately 9.8×10^7 environment steps. These configurations reflect the respective training regimes; the training budgets are not matched across regimes. We therefore assess environment-step efficiency and wall-clock efficiency separately in Figures 9 and 10 and Table 8.

Findings. Table 8 reports final returns and, for each task, the number of environment steps and elapsed wall-clock time required for the 20-iteration running mean to first reach 90% of PPO's final return. Three observations follow.

First, the off-policy methods reach this threshold with fewer environment steps on all six tasks. FastTD3 and FastSAC require 2.7 and 3.7 million steps, respectively, on walker-run, compared with 13.0 to 23.1 million for the on-policy methods. On cartpole-swingup, the corresponding figures are

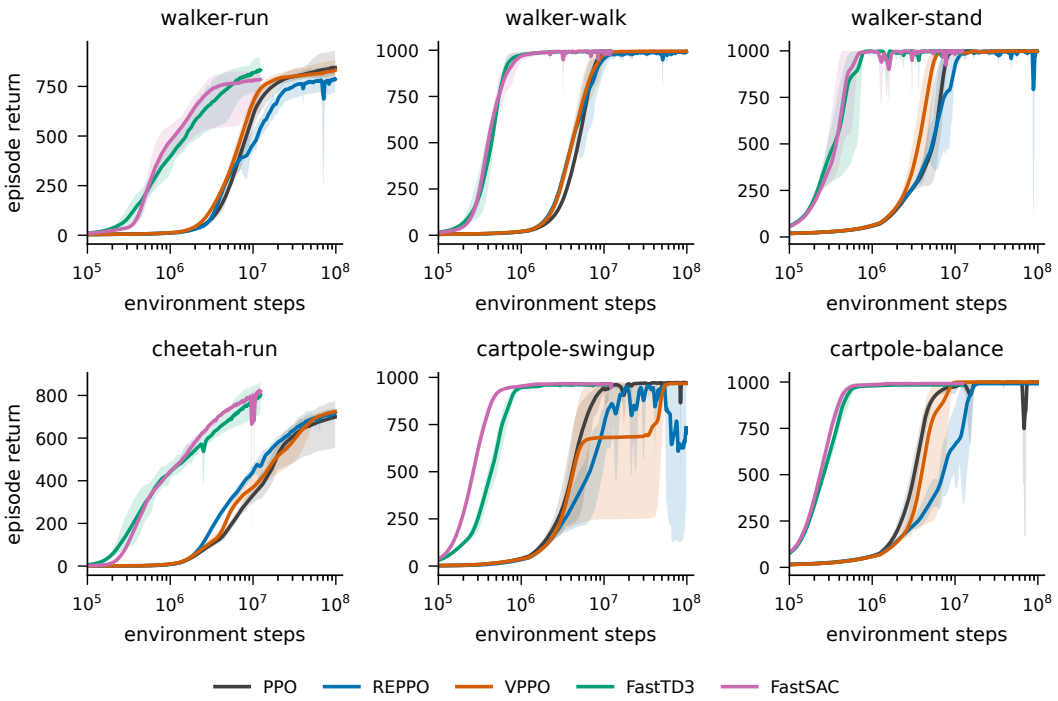


Figure 9: Six DMC-style tasks, return against environment steps (logarithmic axis; 20-iteration running means; bands are seed ranges, five seeds per method). The off-policy methods reach the same return with 4 to 14 times fewer environment steps; the on-policy methods run 2048 environments, the off-policy methods 128. On walker-run the VPPO curve is the five-seed configuration of Table 1.

0.5 and 0.7 million, compared with 5.2 to 11.5 million. These results are consistent with the sample efficiency typically associated with off-policy learning and motivate restricting our efficiency claims to comparisons among on-policy methods.

Second, at the reported levels of parallelism, PPO reaches the threshold sooner in wall-clock time than both off-policy methods on every task. VPPO does so on five of the six tasks; the exception is walker-walk, where its threshold-crossing time differs from FastTD3's by less than 10 seconds. VPPO's higher per-iteration cost (Table 4) contributes to its smaller wall-clock advantage over the off-policy methods.

Third, the off-policy methods achieve similar mean final returns to the on-policy methods on the four ceiling tasks and on walker-run (831 and 784, compared with 785 to 845). On cheetah-run, both off-policy methods achieve higher mean final returns than all on-policy methods (818 and 799, compared with 698 to 724). FastTD3's advantage over the best on-policy mean is 94 points, exceeding the largest difference between on-policy means on that task.

Across the two running tasks, standard deviations range from 54 to 81 for the off-policy methods and from 10 to 75 for the on-policy methods. One FastSAC seed on walker-run and one FastTD3 seed on cheetah-run never exceed returns of 622 and 708, respectively. On walker-stand, the FastSAC learning curve shows a temporary decline followed by recovery; no comparable decline is observed for FastTD3 or VPPO.

These comparisons are descriptive and based on five seeds per method. The on-policy and off-policy experiments use different environment counts, training budgets, and seed sets; we make no claims of statistical significance.

G1 flat terrain. On G1 flat terrain, we evaluated the off-policy methods using 256 parallel environments, 24-step rollouts, and 5000 training iterations, corresponding to approximately 3.1×10^7 environment steps, compared with 4.9×10^8 for the on-policy runs. Each off-policy method was evaluated over three seeds. We used the configuration validated on the DMC-style tasks, with only two G1-specific adjustments: a target entropy of $+0.62\, n_A$ nats, corresponding approximately to a

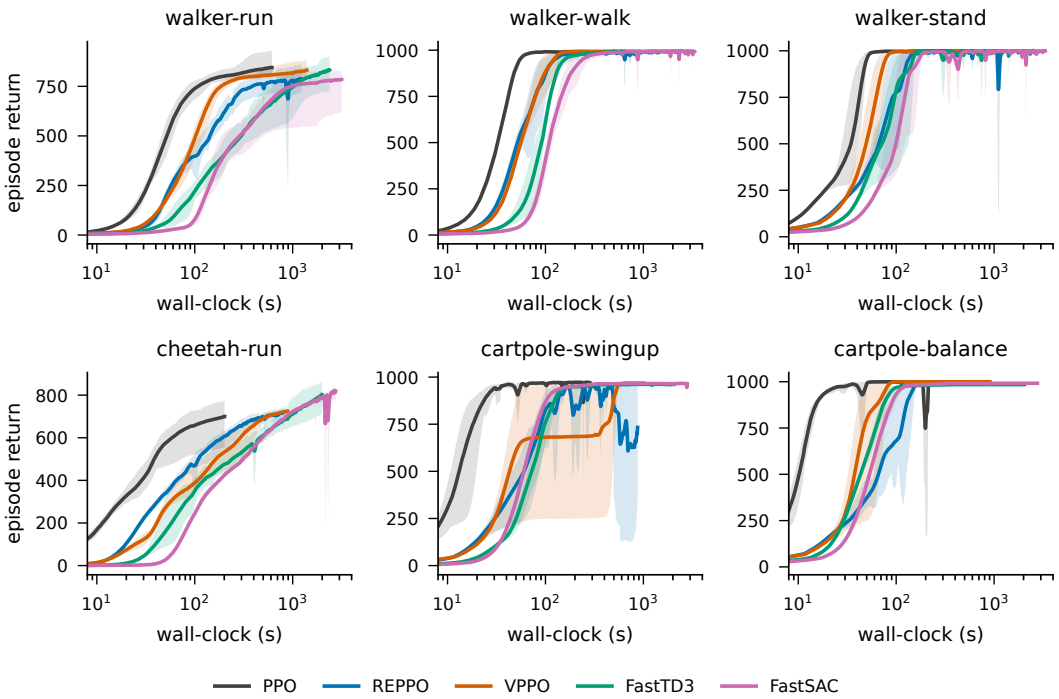


Figure 10: The same runs as Figure 9 against wall-clock time on one GPU (logarithmic axis). At 2048 environments PPO reaches every threshold 3 to 10 times sooner than the off-policy methods and VPPO does so on five of the six tasks.

Gaussian with standard deviation 0.45 in each dimension, and a tanh transformation limiting actor outputs to $[-6, 6]$.

FastSAC achieves a mean final return of 31.2 ± 0.1 (individual seed returns: 31.3, 31.2, and 31.2), with a peak return of 32.5. The corresponding on-policy returns are 81.7 for VPPO and 81.7 ± 0.4 for PPO under the matched-coefficient setting (Table 6). FastSAC requires 8.5×10^3 seconds of training, compared with 8.2×10^3 for VPPO and 3.8×10^3 for PPO.

FastTD3 becomes numerically unstable in all three seeds: critic values become non-finite within the first few training iterations under the configuration with 128 updates per iteration and a batch size of 32768. The trust-gated variant also becomes unstable. Its calibrated weight remains at the floor $q_{\min}$ throughout training, and its critic does not become reliable. Preliminary runs at the on-policy parallelism of 4096 environments, using one seed per method, yielded a final return of 35.9 for FastSAC and a collapse after a peak return of 41.1 for FastTD3.

These results characterize the tested configurations under a limited tuning budget. We did not explore task-specific critic scaling or exploration schedules, so the FastSAC result should not be interpreted as representative of the performance achievable with a fully tuned off-policy baseline. Similarly, the FastTD3 failures establish numerical instability in the tested configurations, rather than an inherent limitation of the algorithm.

H Deterministic Policy Evaluation

Table 10 reports deterministic mean-policy evaluations of the final VPPO checkpoints on flat and rough terrain and the PPO rough-terrain checkpoint trained with the standard configuration (seed 0). All checkpoints are evaluated on the same command grid. These results provide the detailed measurements for the comparison in Section 7.3.

Base height, tracking errors, contact statistics, action rate, and falls are reported. Height is the mean base height over the episode. Velocity and yaw errors are the mean absolute errors in tracking the commanded linear and angular velocities, respectively. Stance fraction denotes the fraction

Table 8: On-policy and off-policy performance on the DMC-style tasks. Final return (mean ± standard deviation over seeds, last 100 iterations; boldface marks the best mean on the task), median environment steps (millions) and median wall-clock seconds at which the 20-iteration running mean first reaches 90% of PPO's final return (the threshold is in parentheses after the task name). A superscript (k/n) marks configurations in which only k of n seeds reached the threshold; the medians are over those seeds. The on-policy methods use 2048 environments and the off-policy methods 128; the walker-run VPPO entry uses five fresh seeds of the final configuration (Table 1), and the REPPO statistics on cartpole-swingup include seeds with performance collapse.

Task (threshold)	Method	Seeds	Final return	Steps to thr. (10^6)	Time to thr. (s)
walker-run (760)	PPO	5	$\mathbf{845 \pm 43}$	16.8	140
	REPPO	5	785 ± 54	$23.1^{(3/5)}$	$300^{(3/5)}$
	VPPO	5	830 ± 29	13.0	335
	FastTD3	5	831 ± 59	$3.7^{(4/5)}$	$1139^{(4/5)}$
	FastSAC	5	784 ± 81	$2.7^{(4/5)}$	$813^{(4/5)}$
walker-walk (894)	PPO	5	993 ± 1	7.6	51
	REPPO	5	989 ± 4	7.9	130
	VPPO	5	994 ± 1	7.1	154
	FastTD3	5	$\mathbf{994 \pm 1}$	0.7	146
	FastSAC	5	993 ± 2	0.7	226
walker-stand (899)	PPO	5	$\mathbf{999 \pm 1}$	7.1	50
	REPPO	5	993 ± 6	10.2	129
	VPPO	5	998 ± 1	4.8	120
	FastTD3	5	$\mathbf{999 \pm 1}$	0.5	144
	FastSAC	5	$\mathbf{999 \pm 1}$	0.5	166
cheetah-run (629)	PPO	5	698 ± 75	$26.9^{(4/5)}$	$56^{(4/5)}$
	REPPO	5	717 ± 18	27.3	231
	VPPO	5	724 ± 10	31.8	298
	FastTD3	5	799 ± 54	3.6	580
	FastSAC	5	$\mathbf{818 \pm 22}$	2.5	566
cartpole-swingup (875)	PPO	5	$\mathbf{972 \pm 3}$	6.1	19
	REPPO	5	689 ± 314	11.5	104
	VPPO	5	968 ± 7	5.2	52
	FastTD3	5	961	0.7	118
	FastSAC	5	961 ± 11	0.5	104
cartpole-balance (900)	PPO	5	**1000**	5.0	15
	REPPO	5	991	13.0	120
	VPPO	5	**1000**	5.4	52
	FastTD3	5	985	0.5	75
	FastSAC	5	992	0.4	91

of time with at least one foot in contact, air time is the mean swing duration, and landing force is the mean peak contact force at touchdown. Action rate is the mean squared change in action between consecutive control steps. Errors and rates use the normalized units recorded by the task implementation. Neither VPPO policy fell under any command in the grid, whereas the PPO rough-terrain policy fell three times under the 1.0 m/s walking command. These fixed-command evaluations complement the training-time fall statistics by characterizing the behavior of the final deterministic policies.

Table 9: Configuration of the off-policy reference points on the DMC-style tasks (both methods share the entries without a method name).

Method	Setting	Value
	environments; rollout length; iterations	128; 32; 3000
	replay length (rows $\times$ environments)	$8192 \times 128 \approx 10^6$ transitions
	gradient updates per iteration; batch size	128; 32768
	random-action warm-up	first 256 rows
	networks (actor and critics); activation	256–128; ELU
	critics; LayerNorm in critic trunks	2; yes
	distributional heads (HL-Gauss); support; bins	yes; $[-110, 110]$; 151
	n-step target; γ; Polyak rate; learning rate	3; 0.99; 0.1; 3×10^{-4}
	observation normalization; action bound	running statistics; $[-1, 1]$ (tanh)
FastTD3	exploration noise; target noise, clip; policy delay	0.2; 0.2, 0.5; 2
FastSAC	target entropy; initial temperature	$-0.5\, n_A$; 0.2
G1 (both)	environments; rollout length; replay rows; networks	256; 24; 4096; 512–256–128
	critics; distributional; action bound; exploration noise (TD3)	2 (scalar, LayerNorm); no; ± 6 (tanh); 0.2
	FastSAC target entropy; initial temperature; temperature learning rate	$+0.62\, n_A$; 0.02; 10^{-3}

Table 10: Deterministic evaluation across the full command grid for the VPPO flat- and rough-terrain checkpoints and the PPO rough-terrain checkpoint trained with the standard configuration.

Checkpoint	Command	Height (m)	Vel. err.	Yaw err.	Stance	Air time (s)	Landing force (N)	Action rate	Falls
VPPO flat	stand	0.761	0.024	0.031	0.994	0.038	297.8	0.032	0
	walk 0.5 m/s	0.753	0.063	0.065	0.663	0.065	162.4	0.102	0
	walk 1.0 m/s	0.742	0.109	0.096	0.612	0.073	223.8	0.143	0
	back 0.5 m/s	0.759	0.107	0.067	0.622	0.068	116.6	0.098	0
	side 0.5 m/s	0.758	0.087	0.050	0.617	0.068	167.0	0.096	0
	turn 0.5 rad/s	0.755	0.071	0.094	0.872	0.038	82.5	0.063	0
VPPO rough	stand	0.759	0.035	0.031	0.990	0.038	273.1	0.013	0
	walk 0.5 m/s	0.782	0.218	0.113	0.727	0.061	141.1	0.096	0
	walk 1.0 m/s	0.722	0.294	0.165	0.602	0.083	233.9	0.147	0
	back 0.5 m/s	0.742	0.241	0.140	0.717	0.058	228.6	0.109	0
	side 0.5 m/s	0.781	0.254	0.138	0.752	0.059	213.4	0.092	0
	turn 0.5 rad/s	0.725	0.049	0.296	0.953	0.030	162.6	0.041	0
PPO rough (standard)	stand	0.751	0.029	0.038	0.991	0.040	255.3	0.022	0
	walk 0.5 m/s	0.775	0.160	0.201	0.698	0.073	174.3	0.087	0
	walk 1.0 m/s	0.706	0.201	0.269	0.628	0.086	257.1	0.126	3
	back 0.5 m/s	0.725	0.163	0.204	0.599	0.102	371.8	0.081	0
	side 0.5 m/s	0.753	0.165	0.192	0.661	0.084	247.6	0.079	0
	turn 0.5 rad/s	0.719	0.033	0.470	0.990	0.048	267.8	0.015	0