LOCAL: LOCAL-CURVATURE ADVANTAGE LOGIT REGRESSION FOR SCALABLE POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Policy optimization should account for how parameter updates change the policy distribution rather than their magnitude in parameter space. Natural policy gradient (NPG) and policy mirror descent (PMD) achieve this through parameter-space and state-wise curvature, respectively, and yield the same Boltzmann policy update for tabular softmax policies. This update determines the target logits only up to a state-dependent offset (SDO), since adding a common constant to all logits leaves the policy unchanged. While therefore negligible in the tabular setting, we show that the SDO becomes consequential under function approximation: fixing it during target fitting alters the curvature beyond the local Fisher geometry. We introduce LOCAL (***Lo**cal-**C**urvature **A**dvantage **L**ogit Regression*), which analytically profiles out the SDO and regresses only the policy-relevant relative logit changes. We show that the resulting objective preserves the local Fisher curvature in logit space and that its local linearization recovers NPG. We further derive a first-order implementation and a Top-K approximation that makes the method practical for very large action spaces. Across Atari and Procgen controls and LLM post-training with RLHF and RLVR, LOCAL consistently improves policy optimization, showing that properly handling this seemingly redundant degree of freedom is important for policy learning with function approximation.

1 INTRODUCTION

Reinforcement learning (RL) provides a general framework for sequential decision making, with applications ranging from visual control (Mnih et al., 2015) to post-training large language models through reinforcement learning from human feedback (Bai et al., 2022, RLHF) and reinforcement learning with verifiable rewards (Shao et al., 2024, RLVR). A central challenge, however, is that the magnitude of a parameter update does not directly characterize the resulting change in the policy distribution. This has motivated curvature-aware policy optimization methods, which measure updates according to their effect on the policy rather than solely in parameter space (Schulman et al., 2015; 2017; Wu et al., 2017). Natural Policy Gradient (Kakade, 2001, NPG) and Policy Mirror Descent (Geist et al., 2019, PMD) provide two closely related views of such curvature-aware optimization. NPG aggregates the local curvature of the policy distribution over the state-action distribution induced by the current policy, yielding a parameter-space Fisher information matrix. PMD instead performs policy improvement directly in distribution space: at each state, it constructs an improved policy by solving a KL-regularized optimization problem, thereby incorporating the local curvature of the policy distribution into the update (Xiao, 2022; Alfano et al., 2023). Although these approaches operate at different levels, they coincide in an important special case. For tabular softmax policies with negative-entropy mirror maps, both NPG and PMD induce the same Boltzmann policy update (Agarwal et al., 2021; Xiao, 2022).

This equivalence reveals a subtle ambiguity that is largely irrelevant in the tabular setting but becomes important under function approximation. In logit space, the tabular update can be represented by adding a scaled advantage to the current logits. Following AMPO (Alfano et al., 2023), we use this representation to fit the policy logits directly. However, the corresponding target logits are not unique: adding the same constant to all action logits at a given state leaves the softmax policy unchanged because the common factor cancels during normalization. Hence, for every state, the target logits are defined only up to an arbitrary state-dependent offset (SDO). Existing analyses typically treat this offset as negligible because it has no effect on the resulting tabular policy (Agarwal et al.,

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.

2021; Alfano et al., 2023). However, under function approximation, this apparently redundant degree of freedom is no longer innocuous. A neural policy cannot generally match all tabular targets exactly, and the choice of SDO changes the regression problem used to fit the policy. In particular, fixing the SDO, for example implicitly setting it to zero, forces the optimizer to match not only the relative logit differences that determine the policy, but also a common logit offset that has no effect on the policy distribution itself. We show that this introduces an additional curvature component into the fitting objective. As a result, the curvature of the regression problem no longer coincides with the local Fisher curvature underlying the corresponding PMD update.

This observation motivates a simple principle: when fitting a curvature-aware policy target, nuisance directions that do not affect the policy should be removed rather than arbitrarily fixed. Based on this principle, we introduce LOCAL (***Lo**cal-**C**urvature **A**dvantage **L**ogit Regression*), a logit-regression formulation that analytically optimizes the SDO for every state during policy fitting. The optimal offset removes the common component of the logit update by centering it under the old policy, so that the regression loss depends only on policy-relevant relative logit changes. We show that the resulting objective preserves the local Fisher curvature of the categorical policy in logit space. Moreover, when the neural policy is locally linearized around the current parameters, the resulting update recovers NPG. Thus, LOCAL provides a direct bridge between state-wise PMD and parameter-space NPG without explicitly constructing or inverting a parameter-space Fisher matrix. We further derive an average-regret bound and show that the optimal SDO minimizes this bound over offset choices for fixed realized logit updates.

LOCAL is optimized with standard minibatch first-order updates, recomputing the optimal SDO from the current logits at each step. For large action spaces, where exact SDO computation requires storing full old-policy logits or probabilities, we introduce a Top-K approximation using only the most probable actions. Our analysis shows that Top-K affects only the fitting-error term in this bound, with the additional error controlled by the omitted old-policy mass and the normalized logit-displacement range. We evaluate LOCAL on Atari (Bellemare et al., 2013), Procgen (Cobbe et al., 2020), and LLM post-training with RLHF for helpful and harmless dialogue and RLVR for mathematical reasoning. Across these settings, optimizing the SDO improves policy optimization, demonstrating the practical importance of preserving the underlying local curvature.

Our contributions are as follows.

- We introduce LOCAL, which analytically profiles out the SDO, and show that the resulting logit-regression objective preserves the local Fisher curvature of the policy. We further establish that the optimal SDO minimizes the regret upper bound over offset choices for fixed realized logit updates.
- We develop a practical minibatch implementation, together with a Top-K approximation that substantially reduces the storage cost for large action spaces. We show that Top-K affects only the fitting-error term in the regret bound, with its contribution controlled by the omitted old-policy mass.
- We demonstrate the effectiveness of LOCAL across Atari, Procgen, RLHF, and RLVR, showing the importance of optimizing the SDO in practice.

2 PRELIMINARIES

We consider a discounted Markov decision process with bounded rewards $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho_0)$. Let $J(\pi)$ denote the unnormalized discounted return from the initial distribution ρ_0, $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$ the advantage, and $d^\pi_{\rho_0}$ the normalized discounted state-visitation distribution. Full definitions are given in Appendix A.1. For discrete actions, we use the softmax policy $\pi(a|s) = \frac{\exp(z^s(a))}{\sum_{b\in\mathcal{A}}\exp(z^s(b))}$, with logit vector $\boldsymbol{z}^s \in \mathbb{R}^{|\mathcal{A}|}$ and probability vector $\boldsymbol{\pi}^s = (\pi(a|s))_{a\in\mathcal{A}}$. For a differentiable policy $\pi_{\boldsymbol{\theta}}$, policy gradient (PG) (Sutton et al., 1999) follows $\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}})$ in the Euclidean geometry of parameter space $\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}}) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_{\boldsymbol{\theta}}}_{\rho_0},\, a\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[A^{\pi_{\boldsymbol{\theta}}}(s,a)\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}(a|s)\right]$. NPG instead accounts for how parameter changes affect the policy distribution, using the direction $\boldsymbol{F}^\dagger_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}})$ (Kakade, 2001), where $\boldsymbol{F}_{\boldsymbol{\theta}} = \mathbb{E}_{s\sim d^{\pi_{\boldsymbol{\theta}}}_{\rho_0},\, a\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}(a|s)\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}(a|s)^\top\right]$ is the parameter-space Fisher and $\dagger$ denotes the Moore-Penrose pseudoinverse. For tabular softmax policies, NPG induces the same update as the state-wise PMD (Xiao, 2022; Alfano et al., 2023):

Definition 1 (State-wise policy mirror descent)**.** For a step size $\eta > 0$, define

$$\boldsymbol{q}^\star(\cdot|s) = \arg\max_{\boldsymbol{q}\in\Delta(\mathcal{A})} \left\{ \langle \boldsymbol{q}, A^\pi(s,\cdot)\rangle - \frac{1}{\eta} D_{\mathrm{KL}}\left(\boldsymbol{q} \,\|\, \boldsymbol{\pi}^s\right) \right\}. \tag{1}$$

For a softmax policy, the following lemma yields the update rule.

Lemma 1 (State-wise update in softmax logit coordinates (Xiao, 2022; Agarwal et al., 2021; Alfano et al., 2023))**.** *For a fixed state s, let $\boldsymbol{q}^\star(\cdot|s)$ be the solution of Equation (1). For any softmax logit vector $\boldsymbol{z}^{s,\star}$ representing $\boldsymbol{q}^\star(\cdot|s)$, there exists $c(s) \in \mathbb{R}$ such that*

$$\boldsymbol{z}^{s,\star} = \boldsymbol{z}^s + \eta A^\pi(s,\cdot) + c(s)\mathbf{1}, \qquad \boldsymbol{q}^\star(\cdot|s) = \frac{\exp(\boldsymbol{z}^{s,\star})}{\sum_{b\in\mathcal{A}} \exp(z^{s,\star}(b))}. \tag{2}$$

We call $c(s)$ a state-dependent offset *(SDO), where $\mathbf{1} \in \mathbb{R}^{|\mathcal{A}|}$ is the all-ones vector. Because $c(s)\mathbf{1}$ adds the same scalar to every action logit at state s, it is canceled by softmax normalization and therefore does not change $\boldsymbol{q}^\star(\cdot|s)$.*

3 LOCAL: LOCAL-CURVATURE ADVANTAGE LOGIT REGRESSION

3.1 PROFILING THE STATE-DEPENDENT OFFSET

State-wise policy mirror descent fixes a unique target policy but not a unique set of target logits. At update k, let π_k have logits $\boldsymbol{z}_k$, and write $d_k := d_{\rho_0}^{\pi_k}$. By Lemma 1, all logits representing $\boldsymbol{q}_k^\star(\cdot|s)$ form the equivalence class $\mathcal{T}_k(s) = \{\boldsymbol{z}_k^s + \eta A^{\pi_k}(s,\cdot) + c(s)\mathbf{1} : c(s) \in \mathbb{R}\}$, where $c(s)$ is a scalar function and $\mathbf{1}$ is the all-ones vector. Every member of $\mathcal{T}_k(s)$ represents the same target policy. Let $\mathcal{Z}$ denote a restricted class of functions $\boldsymbol{z} : \mathcal{S} \to \mathbb{R}^{|\mathcal{A}|}$. The joint fitting problem over $\boldsymbol{z} \in \mathcal{Z}$ is

$$\min_{\boldsymbol{z}\in\mathcal{Z}} \min_{c:\mathcal{S}\to\mathbb{R}} \frac{1}{2}\mathbb{E}_{s\sim d_k}\mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[\left(z^s(a) - z_k^s(a) - \eta A^{\pi_k}(s,a) - c(s)\right)^2\right]. \tag{3}$$

We seek to find the optimal c as given below.

Proposition 1 (Optimal state-dependent offset)**.** *Fix a state s and a policy $\pi_k(\cdot|s)$. For a fixed candidate logit function $\boldsymbol{z} \in \mathcal{Z}$, the unique minimizer in Equation (3) is*

$$c_{\boldsymbol{z}}^\star(s) = \mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[z^s(a) - z_k^s(a) - \eta A^{\pi_k}(s,a)\right] = \mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[z^s(a) - z_k^s(a)\right]. \tag{4}$$

The proof is provided in Appendix A.2.

Therefore, profiling out the state-dependent offset yields the following objective

$$\boxed{\mathcal{L}_k(\boldsymbol{z}) = \frac{1}{2}\mathbb{E}_{s\sim d_k}\mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[\left(z^s(a) - z_k^s(a) - c_{\boldsymbol{z}}^\star(s) - \eta A^{\pi_k}(s,a)\right)^2\right]}, \quad \boldsymbol{z} \in \mathcal{Z}. \tag{5}$$

3.2 ROLE OF THE OPTIMAL OFFSET: A LOCAL GEOMETRY VIEW

The choice of the SDO does not affect an exactly represented softmax policy, but it affects the objective used to fit a restricted logit function class. To make this distinction explicit, define the state-wise logit residual as $\boldsymbol{r}_k^s(\boldsymbol{z}) = (r_k^s(a;\boldsymbol{z}))_{a\in\mathcal{A}} = \boldsymbol{z}^s - \boldsymbol{z}_k^s - \eta A^{\pi_k}(s,\cdot)$. We compare the optimal offset $c_{\boldsymbol{z}}(s) = c_{\boldsymbol{z}}^\star(s)$ with the zero offset $c_{\boldsymbol{z}}(s) = 0$ and uniform centering $c_{\boldsymbol{z}}(s) = |\mathcal{A}|^{-1}\sum_{b\in\mathcal{A}} r_k^s(b;\boldsymbol{z})$ in the least-squares problem in Equation (3):

$$\min_{\boldsymbol{z}\in\mathcal{Z}} \frac{1}{2}\mathbb{E}_{s\sim d_k}\mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[\left(r_k^s(a;\boldsymbol{z}) - c_{\boldsymbol{z}}(s)\right)^2\right], \quad c_{\boldsymbol{z}}(s) \in \left\{c_{\boldsymbol{z}}^\star(s),\, 0,\, \frac{1}{|\mathcal{A}|}\sum_{b\in\mathcal{A}} r_k^s(b;\boldsymbol{z})\right\}. \tag{6}$$

Proposition 2 (Offset-induced residual geometries)**.** *For every state s, let $\boldsymbol{F}_s := \mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[\nabla_{\boldsymbol{z}_k^s}\log\pi_k(a|s)\nabla_{\boldsymbol{z}_k^s}\log\pi_k(a|s)^\top\right] = \mathrm{diag}(\boldsymbol{\pi}_k^s) - \boldsymbol{\pi}_k^s\boldsymbol{\pi}_k^{s\top}$ denote the state-wise Fisher matrix in logit coordinates. For every logit residual $\boldsymbol{r}_k^s(\boldsymbol{z})$, the per-state least-squares terms in Equation (6) under the optimal SDO, zero offset, and uniform centering are:*

$$\begin{aligned} &\text{optimal:} \quad \frac{1}{2}\boldsymbol{r}_k^s(\boldsymbol{z})^\top \boldsymbol{F}_s \boldsymbol{r}_k^s(\boldsymbol{z}), \qquad \text{zero:} \quad \frac{1}{2}\boldsymbol{r}_k^s(\boldsymbol{z})^\top \left[\boldsymbol{F}_s + \boldsymbol{\pi}_k^s\boldsymbol{\pi}_k^{s\top}\right] \boldsymbol{r}_k^s(\boldsymbol{z}), \\ &\text{uniform:} \quad \frac{1}{2}\boldsymbol{r}_k^s(\boldsymbol{z})^\top \left[\boldsymbol{F}_s + \left(\boldsymbol{\pi}_k^s - \frac{1}{|\mathcal{A}|}\mathbf{1}\right)\left(\boldsymbol{\pi}_k^s - \frac{1}{|\mathcal{A}|}\mathbf{1}\right)^\top\right] \boldsymbol{r}_k^s(\boldsymbol{z}). \end{aligned} \tag{7}$$

The proof is provided in Appendix A.3.

The identities in Proposition 2 hold state-wise and therefore remain valid when the state-wise losses are averaged under any state distribution or over a finite collection of sampled states. They provide a geometric interpretation of the fitting objective: with the optimal SDO, the approximation error of the logit displacement is measured by the state-wise Fisher seminorm $\frac{1}{2}\|\boldsymbol{r}_k^s(\boldsymbol{z})\|_{\boldsymbol{F}_s}^2$. In contrast, zero offset and uniform centering introduce additional positive-semidefinite terms into the local curvature and generally do not preserve this Fisher geometry exactly. Optimizing the SDO therefore ensures that the approximation error is evaluated under the state-wise Fisher geometry. This motivates the name ***Lo**cal-**C**urvature **A**dvantage **L**ogit Regression* (LOCAL).

Example 1 — Offset rules under a restricted logit class
Consider one state with $\boldsymbol{\pi}_k^s = (0.8, 0.15, 0.05)$, $A^{\pi_k}(s,\cdot) = (0.35, -1.65, -0.65)$, and $\eta = 1$, and restrict the feasible logit displacements to $\mathcal{Z} := \{\boldsymbol{z} : \boldsymbol{z}^s - \boldsymbol{z}_k^s = (\alpha, 0, 0),\ \alpha \in \mathbb{R}\}$. Since $\mathcal{Z}$ is not closed under arbitrary SDOs, the three objectives in Proposition 2 select different candidate logits. Their resulting Fisher residuals are $\frac{3}{80}$ (optimal) $< \frac{5119}{115600}$ (uniform) $\ll \frac{3511}{10000}$ (zero). Negative-entropy AMPO (Alfano et al., 2023) fits the fixed representative $Q^{\pi_k}(s,\cdot) = A^{\pi_k}(s,\cdot) + V^{\pi_k}(s)$ and produces the residual $\frac{3}{80} + \frac{4}{25}(V^{\pi_k}(s) - \frac{7}{5})^2$. Profiling the SDO instead gives the Fisher projection and removes fitting error caused only by the selected equivalent logit representative.

3.3 Neural Reparameterization

The logit-space construction extends to any differentiable neural parameterization. Let $\Theta \subseteq \mathbb{R}^p$ and let $g : \Theta \to (\mathbb{R}^{|\mathcal{A}|})^{\mathcal{S}}$ map shared parameters to logits, $g(\boldsymbol{\theta}) = \boldsymbol{z}_{\boldsymbol{\theta}}$; write $\boldsymbol{z}_k = \boldsymbol{z}_{\boldsymbol{\theta}_k}$, $\pi_k = \pi_{\boldsymbol{\theta}_k}$, $\boldsymbol{\pi}_k^s = \boldsymbol{\pi}_{\boldsymbol{\theta}_k}^s$, and $d_k := d_{\rho_0}^{\pi_{\boldsymbol{\theta}_k}}$. At the population level, we restrict the feasible class in Equation (5) to $g(\Theta)$ and minimize $\mathcal{L}_k(g(\boldsymbol{\theta}))$ over $\boldsymbol{\theta} \in \Theta$.

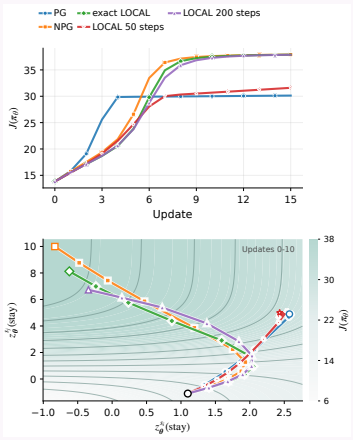
Local connection with NPG. We now show the above optimal SDO is inherently connected to NPG. Let $\boldsymbol{J}_k^s := \partial \boldsymbol{z}_{\boldsymbol{\theta}}^s / \partial \boldsymbol{\theta}^\top \big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_k}$ and linearize the logit displacement as $\boldsymbol{z}_{\boldsymbol{\theta}_k+\Delta\boldsymbol{\theta}}^s - \boldsymbol{z}_{\boldsymbol{\theta}_k}^s \approx \boldsymbol{J}_k^s \Delta\boldsymbol{\theta}$. By Proposition 2, the quadratic term of the linearized optimal SDO objective is

$$\frac{1}{2}\mathbb{E}_{s\sim d_k}\left[(\boldsymbol{J}_k^s\Delta\boldsymbol{\theta})^\top \boldsymbol{F}_s(\boldsymbol{J}_k^s\Delta\boldsymbol{\theta})\right] = \frac{1}{2}\Delta\boldsymbol{\theta}^\top \mathbb{E}_{s\sim d_k}\left[\boldsymbol{J}_k^{s\top}\boldsymbol{F}_s\boldsymbol{J}_k^s\right]\Delta\boldsymbol{\theta}. \tag{8}$$

The matrix above is the parameter-space Fisher obtained by pulling back the state-wise Fisher through the logit map. Together with the policy gradient linear term $\boldsymbol{g}_k := \mathbb{E}_{s\sim d_k}[\boldsymbol{J}_k^{s\top}\boldsymbol{F}_s A^{\pi_k}(s,\cdot)]$, its minimum-norm solution is the NPG direction $\Delta\boldsymbol{\theta}_{\mathrm{NPG}} = \eta \boldsymbol{F}_{\boldsymbol{\theta},k}^\dagger \boldsymbol{g}_k$, where $\boldsymbol{F}_{\boldsymbol{\theta},k} := \mathbb{E}_{s\sim d_k}[\boldsymbol{J}_k^{s\top}\boldsymbol{F}_s\boldsymbol{J}_k^s]$. Thus, the locally linearized optimal SDO objective recovers NPG, while LOCAL directly fits the finite logit displacement. Zero offset and uniform centering generally alter this local geometry through their additional terms.

Example 2 — Fitting accuracy under nonlinear shared logits

Consider the two-state MDP of Kakade (2001). Staying at s_i or s_j yields reward 1 or 2, while switching yields 0 and changes the state. The preferred behavior therefore switches at s_i and stays at s_j. Using the shared logit map $g(\boldsymbol{\theta}) = \boldsymbol{z}_{\boldsymbol{\theta}}$, we set $\boldsymbol{z}_{\boldsymbol{\theta}}^{s_i} = (0, \theta_1)^\top$ and $\boldsymbol{z}_{\boldsymbol{\theta}}^{s_j} = (0, \theta_2 + \theta_1^2)^\top$. With the first action logit fixed to zero, the second equals the log-odds of the second action relative to the first. We compare PG, NPG, exact LOCAL, and approximate LOCAL with 50 or 200 fitting steps. At each policy update, exact LOCAL analytically inverts this nonlinear map to match the state-wise target exactly. Approximate LOCAL instead starts from the current parameters and applies gradient descent steps to fit the objective.
Result. As shown in the figure, PG remains in a lower-return region, whereas NPG and exact LOCAL approach the preferred policy. Increasing the fitting budget from 50 to 200 steps moves approximate LOCAL closer to the exact update in both return and policy trajectory. Appendix B.1 provides complete details.



3.4 Theoretical analysis: Regret Bound for Realized Logit Updates

We now show that the optimal SDO gives the smallest member of the corresponding family of L_2-controlled regret upper bounds in terms of the realized population fitting error. Let $\pi^\star$ be a fixed comparator policy and write $d^\star = d_{\rho_0}^{\pi^\star}$. For each iteration k, define the old-policy fitting distribution $\mu_k(s,a) = d_k(s)\pi_k(a|s)$.

Assumption 1 (State-action concentrability)**.** There exists $C_\mu < \infty$ such that for all $k \geq 0$,

$$\mathbb{E}_{(s,a)\sim\mu_k}\left[\left(\frac{d^\star(s)\pi_{k+1}(a|s)}{\mu_k(s,a)}\right)^2\right] \leq C_\mu \quad \text{and} \quad \mathbb{E}_{(s,a)\sim\mu_k}\left[\left(\frac{d^\star(s)\pi^\star(a|s)}{\mu_k(s,a)}\right)^2\right] \leq C_\mu \tag{9}$$

The concentrability coefficient C_μ quantifies how well the old-policy fitting distribution μ_k covers the distributions $d^\star(s)\pi_{\boldsymbol{\theta}_{k+1}}(a \mid s)$ and $d^\star(s)\pi^\star(a \mid s)$. Similar L_2 concentrability assumptions appear in approximate PMD and NPG analyses (Alfano et al., 2023; Cayci et al., 2024; Yuan et al., 2023). As in these works, the coefficient can be large when the fitting distribution assigns little probability to state-action pairs that are important under the target distributions. Although softmax policies assign positive action probabilities, these probabilities may approach zero during training, so this assumption is not automatically satisfied in practice.

For arbitrary SDOs $c_k : \mathcal{S} \to \mathbb{R}$, define

$$\varepsilon^s_{k,c_k}(a) \coloneqq \eta^{-1}\left[z^s_{\boldsymbol{\theta}_{k+1}}(a) - z^s_{\boldsymbol{\theta}_k}(a) - c_k(s)\right] - A^{\pi_k}(s,a), \quad \varepsilon^s_k(a) \coloneqq \varepsilon^s_{k,c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}_{k+1}}}}(a). \tag{10}$$

Summarize the realized population fitting error over T updates by

$$\varepsilon^2_{c,T} \coloneqq \frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}_{\mu_k}\left[\left(\varepsilon^s_{k,c_k}(a)\right)^2\right], \qquad \varepsilon^2_T \coloneqq \varepsilon^2_{c,T}\big|_{c_k(s)=c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}_{k+1}}}(s)} = \frac{2}{\eta^2 T}\sum_{k=0}^{T-1}\mathcal{L}_k(\boldsymbol{z}_{\boldsymbol{\theta}_{k+1}}). \tag{11}$$

Theorem 1 (Optimal SDO minimizes the fitting error term)**.** *Under Assumption 1, let the comparator $\pi^\star$ be an optimal policy, i.e., $\pi^\star \in \arg\max_\pi J(\pi)$. Let $D_0 \coloneqq \mathbb{E}_{s\sim d^\star}\left[D_{\mathrm{KL}}\left(\boldsymbol{\pi}^{s,\star}\|\boldsymbol{\pi}^s_0\right)\right] \in (0,\infty)$. Recall the bounded reward $|r(s,a)| \leq R_{\max}, \forall s \in \mathcal{S}, a \in \mathcal{A}$, the advantages satisfy $|A^{\pi_k}(s,a)| \leq B_A = \frac{2R_{\max}}{1-\gamma}, \forall k,s,a$. With the fixed step size $\eta = \sqrt{2D_0/(B_A^2 T)}$, for every collection of SDOs $\{c_k : \mathcal{S} \to \mathbb{R}\}_{k=0}^{T-1}$ evaluated on the same realized logit displacements,*

$$\frac{1}{T}\sum_{k=0}^{T-1}\left(J(\pi^\star) - J(\pi_k)\right) \leq \mathcal{C}_1 T^{-\frac{1}{2}} + \mathcal{C}_2\varepsilon_T = \min_{\{c_k:\mathcal{S}\to\mathbb{R}\}_{k=0}^{T-1}}\left\{\mathcal{C}_1 T^{-\frac{1}{2}} + \mathcal{C}_2\varepsilon_{c,T}\right\}. \tag{12}$$

where $\mathcal{C}_1 \coloneqq B_A\sqrt{2D_0}/(1-\gamma)$ and $\mathcal{C}_2 \coloneqq 2\sqrt{C_\mu}/(1-\gamma)$.
Moreover, $\varepsilon^2_{c,T} = \varepsilon^2_T + \frac{1}{\eta^2 T}\sum_{k=0}^{T-1}\mathbb{E}_{s\sim d_k}\left[\left(c_k(s) - c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}_{k+1}}}(s)\right)^2\right]$.

The proof is in Appendix A.4. The first term in Equation (12) is the usual $O(T^{-1/2})$ optimization term, while the second term depends on the realized population fitting error. For the same realized logit displacements, the decomposition in Theorem 1 shows that any SDO other than $c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}_{k+1}}}(s)$ adds a nonnegative squared error. The optimal SDO removes this action-independent component and consequently gives the smallest L_2-based regret upper bound for the realized policy sequence.

3.5 Practical Algorithm and Top-K Approximation for Large Action Spaces

Practical Objective. At iteration k, we roll out the current policy π_k to collect a batch of state–action pairs $\mathcal{D}_k = \{(s_t,a_t)\}_{t=1}^N$ and estimate their advantages $\{\widehat{A}_t\}_{t=1}^N$. For a sampled state–action pair (s_t,a_t) and a candidate logit function $\boldsymbol{z}_{\boldsymbol{\theta}} \in \mathcal{Z}$, define the optimal SDO $c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}}}(s_t) \coloneqq \sum_{b\in\mathcal{A}}\pi_k(b|s_t)\left[z^{s_t}_{\boldsymbol{\theta}}(b) - z^{s_t}_{\boldsymbol{\theta}_k}(b)\right]$. Given a minibatch $\mathcal{B} \subseteq \mathcal{D}_k$, the empirical actor loss is

$$\widehat{\mathcal{L}}_k(\boldsymbol{\theta}) = \frac{1}{2|\mathcal{B}|}\sum_{t\in\mathcal{B}}\left(z^{s_t}_{\boldsymbol{\theta}}(a_t) - z^{s_t}_{\boldsymbol{\theta}_k}(a_t) - c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}}}(s_t) - \eta\widehat{A}_t\right)^2, \qquad \boldsymbol{\theta} \in \Theta. \tag{13}$$

The old-policy logits, probabilities, and advantage estimates are held fixed throughout the update. We optimize Equation (13) for a fixed number of minibatches and use the resulting logit as $\boldsymbol{z}_{\boldsymbol{\theta}_{k+1}}$.

Top-K approximation for large action spaces. When the action space contains tens of thousands of tokens, storing the full old-policy logit vector or probability distribution at every generated token can therefore introduce substantial memory overhead. In LLM post-training, the next-token distribution of an LLM is often concentrated on a relatively small subset of the vocabulary. We use this property to approximate the optimal SDO using only the K tokens with the largest probabilities under the old policy. For each sampled state s_t, let $\mathcal{T}_t^K = \mathrm{TopK}(\boldsymbol{\pi}_k^{s_t}, K), 1 \leq K \leq |\mathcal{A}|$, denote the old-policy Top-K support. We normalize the retained probabilities as $\overline{\pi}_k^K(b|s_t) = \pi_k(b|s_t) / \sum_{j \in \mathcal{T}_t^K} \pi_k(j|s_t)$ for $b \in \mathcal{T}_t^K$. Define the approximated optimal SDO $c_{\boldsymbol{z}_{\boldsymbol{\theta}}}^K(s_t) := \sum_{b \in \mathcal{T}_t^K} \overline{\pi}_k^K(b|s_t) \left[z_{\boldsymbol{\theta}}^{s_t}(b) - z_{\boldsymbol{\theta}_k}^{s_t}(b) \right]$. By replacing $c_{\boldsymbol{z}_{\boldsymbol{\theta}}(s_t)}^\star$ in Equation (13) with $c_{\boldsymbol{z}_{\boldsymbol{\theta}}}^K(s_t)$, we obtain the practical Top-K LOCAL objective. Corollary 1 in Appendix A.5 shows that, for fixed realized logit displacements, Top-K adds exactly its normalized mean-squared SDO error to squared fitting residual. The error is bounded by the average squared product of omitted policy mass and the range of normalized logit displacements, and changes only the residual-dependent regret term.

4 EXPERIMENTAL RESULTS

We evaluate LOCAL in two representative regimes with softmax policies: discrete controls with convolutional networks, and large language model post-training with Transformer architectures.

4.1 DISCRETE CONTROLS

Experiment Setup. We first evaluate LOCAL on Atari (Bellemare et al., 2013) and Procgen (Cobbe et al., 2020), comparing LOCAL with PPO (Schulman et al., 2017), ACKTR (Wu et al., 2017), V-MPO (Song et al., 2020) and MDPO (Tomar et al., 2022). Details are in Appendix B.2.

	Atari HNS			Procgen normalized score		
Algorithm	IQM ($\pm$ bootstrap SE)	Mean ($\pm$ seed SD)	Median ($\pm$ bootstrap SE)	IQM ($\pm$ bootstrap SE)	Mean ($\pm$ seed SD)	Median ($\pm$ bootstrap SE)
LOCAL	**151.39**$^{\pm 5.53}$	**812.96**$^{\pm 177.62}$	**141.53**$^{\pm 4.83}$	**42.48**$^{\pm 2.69}$	**43.56**$^{\pm 4.75}$	**39.12**$^{\pm 3.68}$
V-MPO	45.86$^{\pm 2.08}$	226.79$^{\pm 7.43}$	34.50$^{\pm 3.55}$	5.87$^{\pm 1.76}$	3.57$^{\pm 8.13}$	7.61$^{\pm 2.93}$
ACKTR	74.24$^{\pm 3.64}$	296.79$^{\pm 18.12}$	77.14$^{\pm 7.24}$	5.15$^{\pm 1.42}$	11.09$^{\pm 10.51}$	3.83$^{\pm 1.29}$
PPO	104.19$^{\pm 3.15}$	322.49$^{\pm 30.01}$	93.03$^{\pm 9.40}$	26.00$^{\pm 3.40}$	34.25$^{\pm 7.03}$	18.80$^{\pm 4.23}$
MDPO	105.12$^{\pm 4.84}$	426.62$^{\pm 52.38}$	86.49$^{\pm 7.97}$	6.23$^{\pm 1.02}$	11.21$^{\pm 2.89}$	5.37$^{\pm 1.70}$

Table 1: Aggregate normalized performance on Atari and Procgen over 5 random seeds. IQM and median superscripts are bootstrap standard errors obtained by resampling observed seeds within each fixed environment (200,000 replicates).

Atari Games. As shown in Table 1, LOCAL achieves the highest IQM, mean, and median Atari HNS, outperforming the strongest baseline for each metric by 44.0%, 90.6%, and 52.1%, respectively. Specifically, MDPO is the strongest baseline in terms of IQM and mean, while PPO has the strongest baseline median. Accordingly, the seed standard deviation of the mean for LOCAL is noticeably larger than those of the baselines. This variability is highly concentrated in `VideoPinball` and `Gopher`: excluding these two environments reduces the standard deviation to 18.74 (Atari Results without `VideoPinball` and `Gopher` are provided in Table 12). Figures 4 and 5 and Table 11 provide the learning curves and per-environment HNS results. Our baselines are also competitive with prior implementations. Related Atari results are reported in Le et al. (2022); Huang et al. (2022); Wu et al. (2017); Schulman et al. (2017).

Procgen. As shown in Table 1, LOCAL outperforms the strongest baseline PPO on all three aggregate metrics, including an IQM increase from 26.00 to 42.48, with lower variability (Table 1). Per-environment learning curves are in Figure 6.

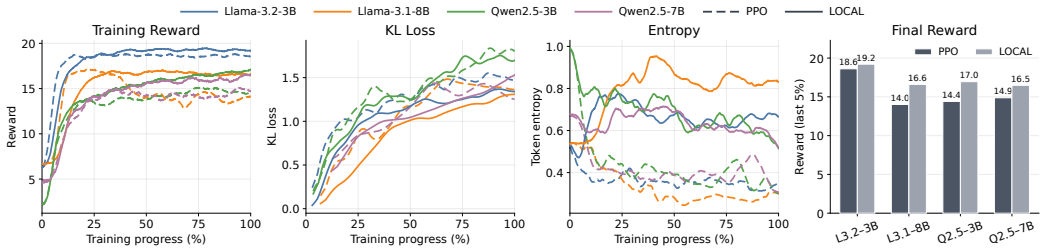


Figure 1: RLHF training results across four Llama and Qwen models. We report all RLHF training diagnostics in Figure 10.

4.2 Large Language Model Post-Training

Experiment Setup. We next evaluate LOCAL with autoregressive language models, where the softmax action space corresponds to the model vocabulary. We conduct RLHF and RLVR experiments with Qwen (Yang et al., 2024a) and Llama models (Team, 2024) ranging from 3B to 8B parameters. For RLHF, we compare standard PPO with LOCAL using the same GAE advantages. For RLVR, we perform two paired comparisons: PPO with LOCAL using the same GAE advantages, and GRPO with LOCAL using group-relative advantages. Across all comparisons, we keep the model initialization, rollout settings, reward signal, advantage estimator, optimizer, and training budget fixed, and change only the actor objective. The detailed configuration is in Appendix B.3.

RLHF and RLVR. Figure 1 summarizes the RLHF results across four Llama and Qwen models. LOCAL consistently achieves higher training reward than PPO while maintaining a similar scale of KL divergence. Over four different models, LOCAL improves the final reward from 18.6 to 19.2, 14.0 to 16.6, 14.4 to 17.0, and 14.9 to 16.5. Table 2 suggests a potential improvement in Math Avg, with observed gains of 0.09–3.97 points in single-seed comparisons. General Avg improves in seven of the eight comparisons, with the only decrease for Qwen2.5-7B with group-relative advantages. These results indicate that LOCAL improves mathematical reasoning across the tested settings while generally preserving the models' broader capabilities.

KL loss and Entropy. KL loss and token entropy are important stability diagnostics in LLM post-training. In Figures 1 and 10 to 12, we report the KL loss as the token average of the sampled-action estimate of $D_{\mathrm{KL}}(\pi_{\boldsymbol{\theta}}(\cdot|s)\|\pi_{\mathrm{ref}}(\cdot|s))$, over response-token states s, where π_{ref} is a frozen copy of the actor at the start of training, and report entropy as the corresponding token average of the full-vocabulary conditional entropy $\mathcal{H}(\pi_{\boldsymbol{\theta}}(\cdot|s)) := -\sum_{a\in\mathcal{A}} \pi_{\boldsymbol{\theta}}(a|s)\log\pi_{\boldsymbol{\theta}}(a|s)$. At the end of training, LOCAL achieves higher reward or evaluation performance with lower KL loss in most displayed comparisons. It also has higher entropy than PPO or GRPO in most comparisons. The RLHF entropy trajectories reveal a further qualitative difference between the methods. Entropy under PPO declines across all four models. For the two Llama models, however, entropy under LOCAL rises after a brief initial transient before settling into a relatively stable plateau.

4.3 Ablations and Sensitivity Analysis

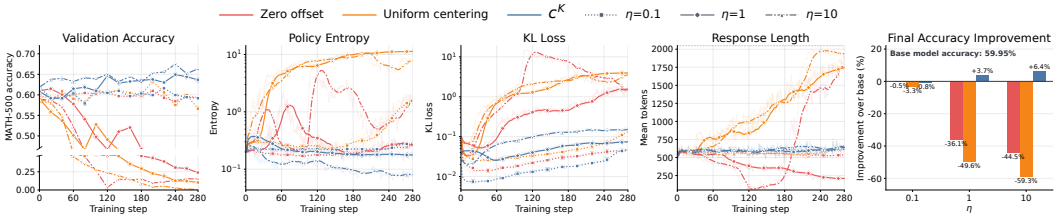


Figure 2: SDO ablation for RLVR with Qwen2.5-3B-Instruct. The model is evaluated on the same fixed 10% subset of MATH-500 .

Model	Adv.	Method	Mathematical Reasoning							General Capabilities				
			MATH 500	MATH Hard	GSM 8K	GSM P1	GSM P2	AMC 23	Avg.	GPQA	MMLU Pro	HotpotQA EM	HotpotQA F1	Avg.
Llama-3.2-3B	Base	Base	47.35	26.02	76.18	54.85	34.14	46.56	47.52	17.38	25.21	20.26	28.01	22.72
	PPO	PPO	53.95	28.74	80.70	**63.19**	41.20	45.31	52.18	24.47	**30.72**	14.98	19.94	22.53
		LOCAL	**56.73**	**34.13**	**81.50**	62.05	**43.12**	**59.38**	**56.15**	**24.94**	28.54	**17.98**	**23.63**	**23.77**
	GRPO	GRPO	**59.45**	33.87	**82.66**	61.11	44.74	52.50	55.72	11.08	31.89	14.07	19.55	19.15
		LOCAL	59.30	**36.84**	82.25	**63.45**	**46.50**	**57.81**	**57.69**	**21.90**	**32.64**	**15.58**	**20.83**	**22.74**
Llama-3.1-8B	Base	Base	56.27	34.43	87.59	73.93	55.68	61.56	61.58	24.83	34.46	45.34	58.60	40.81
	PPO	PPO	59.13	35.81	87.74	74.97	**56.57**	**63.12**	62.89	**25.92**	**36.90**	45.42	58.67	41.73
		LOCAL	**59.48**	**36.02**	**88.02**	**75.02**	56.50	62.81	**62.98**	25.75	36.77	**45.81**	**59.24**	**41.89**
	GRPO	GRPO	**60.43**	37.04	88.25	75.93	57.69	60.94	63.38	26.26	38.35	46.27	59.90	42.70
		LOCAL	60.35	**37.58**	**88.29**	**76.08**	**58.61**	**62.50**	**63.90**	**26.51**	**38.49**	**46.63**	**60.36**	**43.00**
Qwen2.5-3B	Base	Base	67.97	45.19	86.14	73.10	52.18	58.44	63.84	27.82	37.62	27.36	34.27	31.77
	PPO	PPO	69.05	45.54	**86.89**	**74.03**	**53.08**	57.50	64.35	28.77	37.85	**27.83**	**35.01**	32.37
		LOCAL	**69.25**	**45.82**	86.85	73.58	52.68	**59.06**	**64.54**	**30.05**	**37.88**	27.67	34.76	**32.59**
	GRPO	GRPO	71.15	48.28	**88.30**	75.58	53.90	59.06	66.05	28.74	**40.49**	12.96	15.65	24.46
		LOCAL	**71.33**	**49.24**	88.13	**76.24**	**55.27**	**62.19**	**67.07**	**29.38**	39.71	**41.82**	**53.05**	**40.99**
Qwen2.5-7B	Base	Base	75.75	56.30	92.94	84.03	71.81	70.94	75.30	32.73	41.50	49.04	62.42	46.42
	PPO	PPO	**76.62**	**56.88**	93.38	**84.24**	72.07	70.31	75.58	32.39	41.86	49.08	62.46	46.45
		LOCAL	76.45	56.77	**93.39**	84.20	**72.57**	**71.56**	**75.82**	**34.43**	**41.95**	**49.11**	**62.48**	**46.99**
	GRPO	GRPO	76.48	**57.60**	93.37	84.33	72.48	70.94	75.87	**33.15**	40.98	**49.23**	**62.63**	**46.50**
		LOCAL	**76.85**	57.50	**93.51**	**84.49**	**72.69**	**73.12**	**76.36**	32.14	**41.81**	49.18	62.55	46.42

Table 2: RLVR Average@8 evaluation results. Adv. groups results by advantage estimator, while Method compares LOCAL with the corresponding baseline using the same advantages.

State-dependent Offset. As shown in Figure 2, at the conservative step size $\eta = 0.1$, all three variants remain near the base-model accuracy. At the more aggressive step sizes $\eta \in \{1, 10\}$, only the optimal SDO runs avoid accuracy collapse. During later training, both alternatives also have markedly larger KL losses than the optimal SDO, but fail in different ways. Uniform centering shows sustained upward trends in both entropy and response length. Zero offset instead has sharply fluctuating entropy, while response length generally trends downward; for $\eta = 10$, it first falls close to zero and then rebounds rapidly. Further Atari results in Appendix C.3 show the same trend for both advantage and Q-value based targets. In particular, the Q-value target zero offset variant exactly recovers negative-entropy AMPO (Alfano et al., 2023). RLHF results are also reported in Appendix C.4. Across these tasks, zero offset and uniform centering do not reliably support stable and effective learning, highlighting the practical importance of optimizing the SDO.

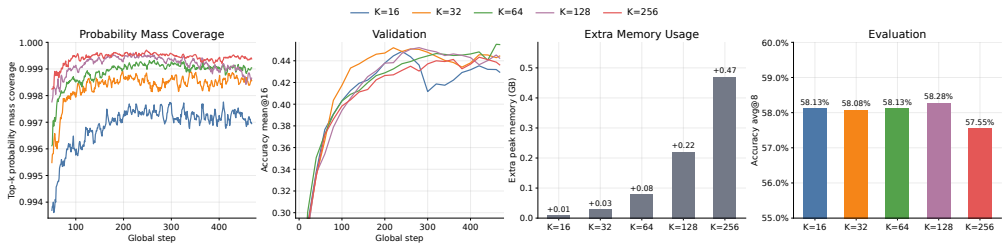


Figure 3: Top-K ($K \in \{16, 32, 64, 128, 256\}$) for RLVR training of Llama-3.2-3B-Instruct. The model is evaluated on the fixed 10% subset of MATH-500.

Top-K Approximation. As shown in Figure 3, even $K = 16$ retains approximately 99.6% of the old-policy probability mass, while increasing K beyond 64 gives only a small absolute increase in coverage. All choices reach a validation mean@16 of approximately 0.43–0.45, with $K = 64$ obtaining the highest final validation accuracy. All final evaluation accuracies are approximately 0.57–0.59, and increasing K does not produce a consistent accuracy improvement. The additional peak memory increases from 0.01 GB for $K = 16$ to 0.47 GB for $K = 256$. This remains much smaller than storing full old-policy logit vectors. More results are deferred to Table 13.

5 Related Work

Policy Mirror Descent. PMD formulates policy improvement as a Bregman-proximal step (Geist et al., 2019; Lan, 2023). For tabular MDPs, Xiao (2022) establishes linear convergence with geometrically increasing step sizes. For regularized problems, Zhan et al. (2023) develop generalized PMD with linear convergence guarantees for convex regularizers. Under function approximation, Sherman et al. (2025) replace restrictive closure conditions with variational gradient dominance and establish convergence rates within the policy class. Practical methods differ in how they realize the policy space update with parameterized policies. MPO and V-MPO use EM-style procedures that construct an improved nonparametric target and fit it through weighted maximum likelihood (Abdolmaleki et al., 2018; Song et al., 2020). MDPO instead approximately solves the proximal objective through multiple gradient steps (Tomar et al., 2022). AMPO fits the dual mirror-descent target by regression, accommodating general score parameterizations and mirror maps (Alfano et al., 2023). PMD-mean regresses log-policy ratios against centered rewards and characterizes the additional χ^2 regularization induced by its approximation of the log-partition term (Xu et al., 2026). *Comparison with PMD methods.* Negative-entropy AMPO regresses toward the fixed Q-based representative $\boldsymbol{z}^s + \eta Q^\pi(s, \cdot)$ without optimizing the SDO (Alfano et al., 2023), which induces the same exact policy as the advantage-based target in LOCAL. LOCAL instead optimizes the SDO and measures the remaining fitting error in the state-wise Fisher geometry without this extra penalty. In contrast to Xu et al. (2026), LOCAL optimizes the SDO to remove the additional offset-induced penalty rather than characterize an additional regularization term, retaining the original PMD target whenever exact fitting is possible.

Natural Policy Gradient. NPG preconditions the policy gradient with the inverse Fisher information matrix in parameter space (Kakade, 2001). Agarwal et al. (2021) establish global convergence guarantees for tabular softmax policies and analyze NPG under function approximation. Cen et al. (2022) establish linear convergence of entropy-regularized tabular NPG and robustness to policy evaluation errors. For log-linear policies, Cayci et al. (2024) prove convergence up to function approximation error. Practical implementations seek to reduce the cost of computing NPG updates. TRPO uses Fisher-vector products and conjugate gradients (Schulman et al., 2015), while ACKTR uses Kronecker-factored curvature approximations (Wu et al., 2017). RAT transforms advantages to estimate NPG using standard backpropagation (Sun, 2026). REBEL fits log-policy-ratio differences to reward differences between paired responses, with its locally linearized population objective recovering an NPG update under independent old-policy sampling (Gao et al., 2024). *Comparison with NPG methods.* LOCAL provides a bridge between PMD and NPG: it fits PMD-induced logit targets, while its locally linearized population objective recovers an NPG step. Under matched state/action formulations, state distributions, and target updates, the REBEL (Gao et al., 2024) and LOCAL objectives coincide in expectation up to a constant scaling factor. With finite action samples, LOCAL uses an analytically computed optimal SDO in place of REBEL's sampled-pair estimation and avoids additional rollouts to form action pairs. Our controlled token-level comparison shows lower curvature-estimation error for both exact and Top-K LOCAL than for sampled pairing, even when the paired estimator uses twice as many action samples (Appendix C.6). Beyond this practical difference, LOCAL provides a clear perspective on how the SDO affects the curvature of PMD-target regression under function approximation and how preserving this geometry connects PMD to NPG.

6 Conclusion and Discussion

We studied how to realize state-wise mirror-descent updates under function approximation, where target logits are defined only up to a state-dependent offset (SDO). Although this freedom leaves the target policy unchanged, its treatment can affect policy fitting. LOCAL analytically optimizes the SDO to preserve the old policy's Fisher geometry in logit space. Its local linearization recovers NPG without constructing or inverting Fisher matrix. Our theoretical results bound average regret through realized fitting error and quantify the effect of Top-K approximation. Experiments on visual control and language-model post-training support the practical importance of optimizing the target-logit representation. Our method also has limitations. First, the theoretical analysis relies on exact advantages, whereas practical training uses noisy estimates. Second, Top-K centering relies on retaining sufficient old-policy probability mass, which may not hold for high-entropy policies.

AI USE STATEMENT

We used generative AI tools to assist with language polishing, grammar and formatting checks, translation, code generation for implementing parts of our method, and identifying related literature. All AI-generated code used in this work was reviewed by the authors. We did not use generative AI tools for generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proofs or assisting with proof writing, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, cleaning or reformatting datasets, supporting qualitative or thematic data analysis, or interpreting results. The authors take full responsibility for the final content of this work, including all text and code produced with AI assistance.

REPRODUCIBILITY STATEMENT

The theoretical assumptions and proofs are provided in Appendix A. Appendix B describes the algorithms, policy parameterizations, experimental settings, and hyperparameters. Additional experimental results and diagnostic analyses are reported in Appendix C.

ETHICS STATEMENT

This work develops a policy optimization method evaluated on simulated control benchmarks and language-model post-training tasks. When applied to language models, policy optimization may retain or amplify biases in the training data, pretrained models, and reward signals. Higher rewards or benchmark scores do not by themselves establish safety or fairness, and our results should be interpreted within the evaluated settings. Downstream applications should assess potential harmful outputs, bias, and unintended behavior arising from imperfect reward signals, while respecting the usage conditions of the underlying datasets and models.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Rémi Munos, Nicolas Heess, and Martin A. Riedmiller. Maximum a posteriori policy optimisation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=S1ANxQW0b.

Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *J. Mach. Learn. Res.*, 22: 98:1–98:76, 2021. URL https://jmlr.org/papers/v22/19-736.html.

Carlo Alfano, Rui Yuan, and Patrick Rebeschini. A novel framework for policy mirror descent with general parameterization and linear convergence. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/61a9278dfef5f871b5e472389f8d6fa1-Abstract-Conference.html.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862, 2022. doi: 10.48550/ARXIV.2204.05862. URL https://doi.org/10.48550/arXiv.2204.05862.

M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279,

2013. ISSN 1076-9757. doi: 10.1613/jair.3912. URL http://dx.doi.org/10.1613/jair.3912.

Semih Cayci, Niao He, and R. Srikant. Convergence of entropy-regularized natural policy gradient with linear function approximation. *SIAM J. Optim.*, 34(3):2729–2755, 2024. doi: 10.1137/22M1540156. URL https://doi.org/10.1137/22m1540156.

Shicong Cen, Chen Cheng, Yuxin Chen, Yuting Wei, and Yuejie Chi. Fast global convergence of natural policy gradient methods with entropy regularization. *Oper. Res.*, 70(4):2563–2578, 2022. doi: 10.1287/OPRE.2021.2151. URL https://doi.org/10.1287/opre.2021.2151.

Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning, 2020. URL https://arxiv.org/abs/1912.01588.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Lasse Espeholt, Hubert Soyer, Rémi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: scalable distributed deep-rl with importance weighted actor-learner architectures. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1406–1415. PMLR, 2018. URL http://proceedings.mlr.press/v80/espeholt18a.html.

Zhaolin Gao, Jonathan Daniel Chang, Wenhao Zhan, Owen Oertell, Gokul Swamy, Kianté Brantley, Thorsten Joachims, J. Andrew Bagnell, Jason D. Lee, and Wen Sun. REBEL: Reinforcement learning via regressing relative rewards. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=yxjWAJzUyV.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized Markov decision processes. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2160–2169. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/geist19a.html.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G. M. Araújo. CleanRL: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022. URL https://jmlr.org/papers/v23/21-1342.html.

Sham M. Kakade. A natural policy gradient. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani (eds.), *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pp. 1531–1538. MIT Press, 2001. URL https://proceedings.neurips.cc/paper/2001/hash/4b86abe48d358ecf194c56c69108433e-Abstract.html.

Sham M. Kakade and John Langford. Approximately optimal approximate reinforcement learning. In Claude Sammut and Achim G. Hoffmann (eds.), *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002*, pp. 267–274. Morgan Kaufmann, 2002.

Guanghui Lan. Policy mirror descent for reinforcement learning: linear convergence, new sampling complexity, and generalized problem classes. *Math. Program.*, 198(1):1059–1106, 2023. doi: 10.1007/S10107-022-01816-5. URL https://doi.org/10.1007/s10107-022-01816-5.

Hung Le, Thommen Karimpanal George, Majid Abdolshah, Dung Nguyen, Kien Do, Sunil Gupta, and Svetha Venkatesh. Learning to constrain policy optimization with virtual trust region. In *Advances in Neural Information Processing Systems*, volume 35, pp. 12775–12786, 2022. doi: 10.52202/068431-0928. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/531998dc1fc858b5857a90b74d96ecab-Paper-Conference.pdf.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=AjXkRZIvjB.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533, 2015. doi: 10.1038/NATURE14236. URL https://doi.org/10.1038/nature14236.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. *CoRR*, abs/2311.12022, 2023. doi: 10.48550/ARXIV.2311.12022. URL https://doi.org/10.48550/arXiv.2311.12022.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/schulman15.html.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Uri Sherman, Tomer Koren, and Yishay Mansour. Convergence of policy mirror descent beyond compatible function approximation. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 54825–54863. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/sherman25a.html.

H. Francis Song, Abbas Abdolmaleki, Jost Tobias Springenberg, Aidan Clark, Hubert Soyer, Jack W. Rae, Seb Noury, Arun Ahuja, Siqi Liu, Dhruva Tirumala, Nicolas Heess, Dan Belov, Martin A. Riedmiller, and Matthew M. Botvinick. V-MPO: on-policy maximum a posteriori policy optimization for discrete and continuous control. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SylOlp4FvH.

Mingfei Sun. Randomized advantage transformation (RAT): Computing natural policy gradients via direct backpropagation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=oVtiySvpXz.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction, 2nd Edition*. MIT Press, 2018. URL http://www.incompleteideas.net/book/the-book-2nd.html.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. Solla, T. Leen, and K. Müller (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=aBO5SvgSt1.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. URL https://arxiv.org/abs/2406.01574.

Yuhuai Wu, Elman Mansimov, Roger Grosse, Shun Liao, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In *Advances in neural information processing systems*, pp. 5279–5288, 2017.

Lin Xiao. On the convergence rates of policy gradient methods. *Journal of Machine Learning Research*, 23(282):1–36, 2022. URL http://jmlr.org/papers/v23/22-0056.html.

Zhenghao Xu, Qin Lu, Changlong Yu, and Tuo Zhao. Approximation of log-partition function in policy mirror descent induces implicit regularization for LLM post-training. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=WPFVuJDDqc.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024a. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *CoRR*, abs/2409.12122, 2024b. doi: 10.48550/ARXIV.2409.12122. URL https://doi.org/10.48550/arXiv.2409.12122.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1259. URL https://doi.org/10.18653/v1/d18-1259.

Rui Yuan, Simon Shaolei Du, Robert M. Gower, Alessandro Lazaric, and Lin Xiao. Linear convergence of natural policy gradient methods with log-linear policies. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=-z9hdsyUwVQ.

Wenhao Zhan, Shicong Cen, Baihe Huang, Yuxin Chen, Jason D. Lee, and Yuejie Chi. Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence. *SIAM J. Optim.*, 33(2):1061–1091, 2023. doi: 10.1137/21M1456789. URL https://doi.org/10.1137/21m1456789.

APPENDIX CONTENTS

A Theoretical Setup and Proofs **16**

A.1 Markov Decision Processes and Performance Difference 16

A.2 Proof of Proposition 1 16

A.3 Proof of Proposition 2 17

A.4 Proof of Theorem 1 18

A.5 Theoretical Analysis: Effect of the Top-K approximation 19

B Experimental Details **20**

B.1 Details of the Two-State Illustrative Experiment 20

B.2 Shared actor-critic implementation for Atari and Procgen 22

B.3 Implementation for LLM post-training 26

C Additional Results **28**

C.1 Atari Games 28

C.2 Procgen 31

C.3 Additional Results for SDO ablation on Atari Games 32

C.4 Additional Results for SDO Ablation in RLHF 33

C.5 Additional Results for Top-K Approximation Ablation 34

C.6 Additional Results on Curvature Estimation Comparison with REBEL (Gao et al., 2024) 34

C.7 Reinforcement Learning with Human Feedback 36

C.8 Reinforcement Learning with Verifiable Rewards 38

A Theoretical Setup and Proofs

A.1 Markov Decision Processes and Performance Difference

We consider a discounted Markov decision process $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho_0)$, where $\mathcal{S}$ is a finite state space, $\mathcal{A}$ is a finite action space, $P(s'|s,a)$ is the transition kernel, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, |r(s,a)| \le R_{\max}, \forall s \in \mathcal{S}, a \in \mathcal{A}$ is a bounded reward function, $\gamma \in (0,1)$ is the discount factor, and ρ_0 is the initial state distribution. A stationary policy is a map $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, where $\Delta(\mathcal{A})$ denotes the probability simplex over $\mathcal{A}$. Starting from $s_0 \sim \rho_0$, the agent samples $a_t \sim \pi(\cdot|s_t)$ and $s_{t+1} \sim P(\cdot|s_t, a_t)$. The objective is to maximize the expected discounted return $J(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$. For a policy π, we define the normalized discounted state visitation distribution as $d^\pi_{\rho_0}(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \Pr^\pi(s_t = s | s_0 \sim \rho_0)$. Let V^π and Q^π denote the standard value and action-value functions (Sutton & Barto, 2018)

$$V^\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\middle|\, s_0 = s\right], \tag{14}$$

$$Q^\pi(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[V^\pi(s')] \tag{15}$$

Define the advantage as $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$, which satisfies $\mathbb{E}_{a\sim\pi(\cdot|s)}[A^\pi(s,a)] = 0, \forall s \in \mathcal{S}$. For any two policies π and π', the performance-difference identity (Kakade & Langford, 2002) gives

$$J(\pi') - J(\pi) = \frac{1}{1-\gamma} \mathbb{E}_{s\sim d^{\pi'}_{\rho_0}, a\sim\pi'(\cdot|s)}\left[A^\pi(s,a)\right]. \tag{16}$$

A.2 Proof of Proposition 1

Proof. Fix a state s and a candidate logit function $\boldsymbol{z} \in \mathcal{Z}$. Define the uncentered logit residual

$$\boldsymbol{r}^s_k(\boldsymbol{z}) = (r^s_k(a;\boldsymbol{z}))_{a\in\mathcal{A}} = \boldsymbol{z}^s - \boldsymbol{z}^s_k - \eta A^{\pi_k}(s,\cdot). \tag{17}$$

For this fixed state, the part of Equation (3) that depends on the SDO $c(s)$ is

$$\ell_s\left(c(s)\right) = \frac{1}{2}\mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[\left(r^s_k(a;\boldsymbol{z}) - c(s)\right)^2\right]. \tag{18}$$

Differentiating with respect to $c(s)$ gives

$$\frac{\partial \ell_s\left(c(s)\right)}{\partial c(s)} = \mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[c(s) - r^s_k(a;\boldsymbol{z})\right] = c(s) - \mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[r^s_k(a;\boldsymbol{z})\right]. \tag{19}$$

The second derivative is $\frac{\partial^2 \ell_s(c(s))}{\partial c(s)^2} = 1$,

so the objective is strictly convex in $c(s)$. Its unique minimizer is therefore

$$c^\star_{\boldsymbol{z}}(s) = \mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[r^s_k(a;\boldsymbol{z})\right] = \mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[z^s(a) - z^s_k(a) - \eta A^{\pi_k}(s,a)\right]. \tag{20}$$

Then, write the optimal SDO as

$$c^\star_{\boldsymbol{z}}(s) = {\boldsymbol{\pi}^s_k}^\top \boldsymbol{r}^s_k(\boldsymbol{z}). \tag{21}$$

It follows that

$$\boldsymbol{r}^s_k(\boldsymbol{z}) - c^\star_{\boldsymbol{z}}(s)\mathbf{1} = \boldsymbol{r}^s_k(\boldsymbol{z}) - \mathbf{1}{\boldsymbol{\pi}^s_k}^\top \boldsymbol{r}^s_k(\boldsymbol{z}) = \left(\boldsymbol{I} - \mathbf{1}{\boldsymbol{\pi}^s_k}^\top\right)\boldsymbol{r}^s_k(\boldsymbol{z}) \tag{22}$$

By the state-wise centering property of the advantage, ${\boldsymbol{\pi}^s_k}^\top A^{\pi_k}(s,\cdot) = 0$,

$$\left(\boldsymbol{I} - \mathbf{1}{\boldsymbol{\pi}^s_k}^\top\right) A^{\pi_k}(s,\cdot) = A^{\pi_k}(s,\cdot). \tag{23}$$

Substituting the definition of $\boldsymbol{r}^s_k(\boldsymbol{z})$ into Equation (22) consequently gives

$$\begin{aligned}
&\boldsymbol{z}^s - \boldsymbol{z}^s_k - \eta A^{\pi_k}(s,\cdot) - c^\star_{\boldsymbol{z}}(s)\mathbf{1} \\
&= \boldsymbol{z}^s - \boldsymbol{z}^s_k - \eta A^{\pi_k}(s,\cdot) - \mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[z^s(a) - z^s_k(a) - \eta A^{\pi_k}(s,a)\right]\mathbf{1} \\
&= \boldsymbol{z}^s - \boldsymbol{z}^s_k - \mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[z^s(a) - z^s_k(a)\right]\mathbf{1} - \eta A^{\pi_k}(s,\cdot).
\end{aligned} \tag{24}$$

This proves the proposition. □

A.3 Proof of Proposition 2

Proof. Fix a state s and an arbitrary candidate logit function $\boldsymbol{z} \in \mathcal{Z}$. Recall the exact logit residual

$$\boldsymbol{r}_k^s(\boldsymbol{z}) = \left(r_k^s(a;\boldsymbol{z})\right)_{a\in\mathcal{A}} = \boldsymbol{z}^s - \boldsymbol{z}_k^s - \eta A^{\pi_k}(s,\cdot). \tag{25}$$

For any SDO $c(s) \in \mathbb{R}$, the state-wise least-squares objective is

$$\frac{1}{2}\mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[\left(r_k^s(a;\boldsymbol{z}) - c(s)\right)^2\right] = \frac{1}{2}\left(\boldsymbol{r}_k^s(\boldsymbol{z}) - c(s)\mathbf{1}\right)^\top \operatorname{diag}\left(\boldsymbol{\pi}_k^s\right)\left(\boldsymbol{r}_k^s(\boldsymbol{z}) - c(s)\mathbf{1}\right). \tag{26}$$

For the optimal SDO, differentiating Equation (26) with respect to $c(s)$ gives

$$0 = -\mathbf{1}^\top \operatorname{diag}\left(\boldsymbol{\pi}_k^s\right)\left(\boldsymbol{r}_k^s(\boldsymbol{z}) - c(s)\mathbf{1}\right) = -{\boldsymbol{\pi}_k^s}^\top \boldsymbol{r}_k^s(\boldsymbol{z}) + c(s), \tag{27}$$

where

$$\mathbf{1}^\top \operatorname{diag}\left(\boldsymbol{\pi}_k^s\right)\mathbf{1} = \sum_{a\in\mathcal{A}} \pi_k(a|s) = 1. \tag{28}$$

Therefore, the optimal SDO is

$$c_{\boldsymbol{z}}^\star(s) = {\boldsymbol{\pi}_k^s}^\top \boldsymbol{r}_k^s(\boldsymbol{z}). \tag{29}$$

Substituting Equation (29) into Equation (26) yields

$$\begin{aligned}\frac{1}{2}\boldsymbol{r}_k^s(\boldsymbol{z})^\top \operatorname{diag}\left(\boldsymbol{\pi}_k^s\right)\boldsymbol{r}_k^s(\boldsymbol{z}) - \frac{1}{2}\left({\boldsymbol{\pi}_k^s}^\top \boldsymbol{r}_k^s(\boldsymbol{z})\right)^2 &= \frac{1}{2}\boldsymbol{r}_k^s(\boldsymbol{z})^\top\left[\operatorname{diag}\left(\boldsymbol{\pi}_k^s\right) - \boldsymbol{\pi}_k^s{\boldsymbol{\pi}_k^s}^\top\right]\boldsymbol{r}_k^s(\boldsymbol{z}) \\ &= \frac{1}{2}\boldsymbol{r}_k^s(\boldsymbol{z})^\top \boldsymbol{F}_s \boldsymbol{r}_k^s(\boldsymbol{z}).\end{aligned} \tag{30}$$

This proves optimal SDO geometry in Equation (7).

For the zero offset, setting $c(s) = 0$ in Equation (26) gives

$$\frac{1}{2}\boldsymbol{r}_k^s(\boldsymbol{z})^\top \operatorname{diag}\left(\boldsymbol{\pi}_k^s\right)\boldsymbol{r}_k^s(\boldsymbol{z}). \tag{31}$$

Using

$$\operatorname{diag}\left(\boldsymbol{\pi}_k^s\right) = \boldsymbol{F}_s + \boldsymbol{\pi}_k^s{\boldsymbol{\pi}_k^s}^\top, \tag{32}$$

we obtain

$$\frac{1}{2}\boldsymbol{r}_k^s(\boldsymbol{z})^\top\left[\boldsymbol{F}_s + \boldsymbol{\pi}_k^s{\boldsymbol{\pi}_k^s}^\top\right]\boldsymbol{r}_k^s(\boldsymbol{z}). \tag{33}$$

This proves zero offset geometry in Equation (7).

For uniform centering, define $\boldsymbol{C}_{\text{uniform}} = \boldsymbol{I} - \frac{1}{|\mathcal{A}|}\mathbf{1}\mathbf{1}^\top$. The uniform centering is

$$c_{\text{uniform}}(s) = \frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} r_k^s(a;\boldsymbol{z}). \tag{34}$$

Consequently,

$$\boldsymbol{r}_k^s(\boldsymbol{z}) - c_{\text{uniform}}(s)\mathbf{1} = \boldsymbol{C}_{\text{uniform}}\boldsymbol{r}_k^s(\boldsymbol{z}). \tag{35}$$

The corresponding state-wise objective is

$$\frac{1}{2}\boldsymbol{r}_k^s(\boldsymbol{z})^\top \boldsymbol{C}_{\text{uniform}}^\top \operatorname{diag}\left(\boldsymbol{\pi}_k^s\right)\boldsymbol{C}_{\text{uniform}}\boldsymbol{r}_k^s(\boldsymbol{z}). \tag{36}$$

Because $\boldsymbol{C}_{\text{uniform}}$ is symmetric, direct expansion gives

$$\boldsymbol{C}_{\text{uniform}}^\top \operatorname{diag}\left(\boldsymbol{\pi}_k^s\right)\boldsymbol{C}_{\text{uniform}} = \operatorname{diag}\left(\boldsymbol{\pi}_k^s\right) - \boldsymbol{\pi}_k^s\frac{\mathbf{1}^\top}{|\mathcal{A}|} - \frac{\mathbf{1}}{|\mathcal{A}|}{\boldsymbol{\pi}_k^s}^\top + \frac{1}{|\mathcal{A}|^2}\mathbf{1}\mathbf{1}^\top. \tag{37}$$

On the other hand,

$$\begin{aligned}
&\boldsymbol{F}_s + \left(\boldsymbol{\pi}_k^s - \frac{1}{|\mathcal{A}|}\mathbf{1}\right)\left(\boldsymbol{\pi}_k^s - \frac{1}{|\mathcal{A}|}\mathbf{1}\right)^\top \\
&= \operatorname{diag}\left(\boldsymbol{\pi}_k^s\right) - \boldsymbol{\pi}_k^s{\boldsymbol{\pi}_k^s}^\top + \boldsymbol{\pi}_k^s{\boldsymbol{\pi}_k^s}^\top - \boldsymbol{\pi}_k^s\frac{\mathbf{1}^\top}{|\mathcal{A}|} - \frac{\mathbf{1}}{|\mathcal{A}|}{\boldsymbol{\pi}_k^s}^\top + \frac{1}{|\mathcal{A}|^2}\mathbf{1}\mathbf{1}^\top \\
&= \operatorname{diag}\left(\boldsymbol{\pi}_k^s\right) - \boldsymbol{\pi}_k^s\frac{\mathbf{1}^\top}{|\mathcal{A}|} - \frac{\mathbf{1}}{|\mathcal{A}|}{\boldsymbol{\pi}_k^s}^\top + \frac{1}{|\mathcal{A}|^2}\mathbf{1}\mathbf{1}^\top. \qquad (38)
\end{aligned}$$

Comparing this expression with Equation (37), we obtain

$$\boldsymbol{C}_{\text{uniform}}^\top \operatorname{diag}\left(\boldsymbol{\pi}_k^s\right)\boldsymbol{C}_{\text{uniform}} = \boldsymbol{F}_s + \left(\boldsymbol{\pi}_k^s - \frac{1}{|\mathcal{A}|}\mathbf{1}\right)\left(\boldsymbol{\pi}_k^s - \frac{1}{|\mathcal{A}|}\mathbf{1}\right)^\top. \qquad (39)$$

Substituting this identity into Equation (36) yields

$$\frac{1}{2}\boldsymbol{r}_k^s(\boldsymbol{z})^\top\left[\boldsymbol{F}_s + \left(\boldsymbol{\pi}_k^s - \frac{1}{|\mathcal{A}|}\mathbf{1}\right)\left(\boldsymbol{\pi}_k^s - \frac{1}{|\mathcal{A}|}\mathbf{1}\right)^\top\right]\boldsymbol{r}_k^s(\boldsymbol{z}). \qquad (40)$$

This proves uniform centering geometry in Equation (7). □

A.4 Proof of Theorem 1

Intermediate Lemma To prove Theorem 1, we need the following lemma:

Lemma 2 (General offset regret inequality)**.** *Suppose that Assumption 1 holds. Let* $D_0 := \mathbb{E}_{s\sim d^\star}\left[D_{\mathrm{KL}}\left(\boldsymbol{\pi}^{s,\star}\|\boldsymbol{\pi}_0^s\right)\right] \in (0,\infty)$*. Recall the bounded reward* $|r(s,a)| \le R_{\max}, \forall s \in \mathcal{S}, a \in \mathcal{A}$*, define the advantage bound* $|A^{\pi_k}(s,a)| \le B_A = \frac{2R_{\max}}{1-\gamma}, \forall k, s, a$*. For any measurable SDOs* $c_k : \mathcal{S} \to \mathbb{R}$ *with finite* $L_2(\mu_k)$ *residuals and any fixed* $\eta > 0$*,*

$$\frac{1}{T}\sum_{k=0}^{T-1}\left(J(\pi^\star) - J(\pi_k)\right) \le \frac{D_0}{(1-\gamma)\eta T} + \frac{\eta B_A^2}{2(1-\gamma)} + \frac{2\sqrt{C_\mu}}{1-\gamma}\varepsilon_{c,T}. \qquad (41)$$

Proof. Recall Equation (10), the definition of $\varepsilon_{k,c_k}^s(a)$ gives

$$z_{\boldsymbol{\theta}_{k+1}}^s(a) - z_{\boldsymbol{\theta}_k}^s(a) = \eta\left(A^{\pi_k}(s,a) + \varepsilon_{k,c_k}^s(a)\right) + c_k(s). \qquad (42)$$

Because $c_k(s)$ is action independent, it cancels under softmax normalization. Hence

$$\pi_{k+1}(a|s) = \frac{\pi_k(a|s)\exp\left(\eta[A^{\pi_k}(s,a) + \varepsilon_{k,c_k}^s(a)]\right)}{Z_{k,c_k}(s)}, \qquad (43)$$

where

$$Z_{k,c_k}(s) := \mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[\exp\left(\eta[A^{\pi_k}(s,a) + \varepsilon_{k,c_k}^s(a)]\right)\right]. \qquad (44)$$

Also define $Z_k^A(s) := \mathbb{E}_{a\sim\pi_k(\cdot|s)}[\exp(\eta A^{\pi_k}(s,a))]$. Comparing Equation (43) with the ideal advantage-tilted policy gives

$$\log Z_{k,c_k}(s) \le \log Z_k^A(s) + \eta\mathbb{E}_{a\sim\pi_{k+1}(\cdot|s)}\left[\varepsilon_{k,c_k}^s(a)\right]. \qquad (45)$$

Since the exact advantage has zero mean under $\pi_k(\cdot|s)$ and is bounded by B_A, Hoeffding's lemma yields

$$\log Z_k^A(s) \le \frac{\eta^2 B_A^2}{2}. \qquad (46)$$

Taking expectation of the log ratio in Equation (43) under $\pi^\star(\cdot|s)$ and applying Equations (45) to (46) gives

$$\begin{aligned}
\mathbb{E}_{a\sim\pi^\star(\cdot|s)}\left[A^{\pi_k}(s,a)\right] \le \frac{1}{\eta}\left[D_{\mathrm{KL}}(\boldsymbol{\pi}^{s,\star}\|\boldsymbol{\pi}_k^s) - D_{\mathrm{KL}}(\boldsymbol{\pi}^{s,\star}\|\boldsymbol{\pi}_{k+1}^s)\right] + \frac{\eta B_A^2}{2} \\
+ \mathbb{E}_{a\sim\pi_{k+1}(\cdot|s)}\left[\varepsilon_{k,c_k}^s(a)\right] - \mathbb{E}_{a\sim\pi^\star(\cdot|s)}\left[\varepsilon_{k,c_k}^s(a)\right]. \qquad (47)
\end{aligned}$$

By the performance-difference identity, the last two expectations in Equation (47) are integrated over $d^\star$. Assumption 1 and Cauchy–Schwarz imply

$$\left|\mathbb{E}_{s\sim d^\star,\, a\sim\pi_{k+1}(\cdot|s)}\left[\varepsilon^s_{k,c_k}(a)\right]\right| \le \sqrt{C_\mu}\left\{\mathbb{E}_{(s,a)\sim\mu_k}\left[\left(\varepsilon^s_{k,c_k}(a)\right)^2\right]\right\}^{1/2},$$
$$\left|\mathbb{E}_{s\sim d^\star,\, a\sim\pi^\star(\cdot|s)}\left[\varepsilon^s_{k,c_k}(a)\right]\right| \le \sqrt{C_\mu}\left\{\mathbb{E}_{(s,a)\sim\mu_k}\left[\left(\varepsilon^s_{k,c_k}(a)\right)^2\right]\right\}^{1/2}. \tag{48}$$

Summing Equation (47) over k, telescoping the KL terms, using Equation (48), and applying Cauchy–Schwarz across iterations prove Equation (41). □

Proof of Theorem 1

Proof. Apply Lemma 2 with $c_k(s) = c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}_{k+1}}}(s)$. The root-mean-square residual in Equation (41) is ε_T by Equation (11). With $\eta = \sqrt{2D_0/(B_A^2 T)}$,

$$\frac{D_0}{(1-\gamma)\eta T} + \frac{\eta B_A^2}{2(1-\gamma)} = \frac{B_A\sqrt{2D_0}}{(1-\gamma)\sqrt{T}}, \tag{49}$$

which proves the first inequality in Equation (12).

Proposition 1 and exact advantage centering imply

$$\mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[\varepsilon^s_k(a)\right] = 0. \tag{50}$$

For any alternative SDO,

$$\varepsilon^s_{k,c_k}(a) = \varepsilon^s_k(a) + \frac{c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}_{k+1}}}(s) - c_k(s)}{\eta}. \tag{51}$$

Expanding the square in Equation (51), using Equation (50), and averaging over states and iterations prove Equation (12). Hence $\varepsilon_{c,T} \ge \varepsilon_T$. □

A.5 Theoretical Analysis: Effect of the Top-K Approximation

We now analyze the effect of the Top-K approximation. Let $\mathcal{T}^K_{k,s}$ be the K actions with the largest probabilities under $\pi_k(\cdot|s)$, and define

$$\alpha^K_k(s) \coloneqq \sum_{a\in\mathcal{T}^K_{k,s}} \pi_k(a|s), \quad c^K_k(s) \coloneqq \sum_{a\in\mathcal{T}^K_{k,s}} \frac{\pi_k(a|s)}{\alpha^K_k(s)}[z^s_{\boldsymbol{\theta}_{k+1}}(a) - z^s_{\boldsymbol{\theta}_k}(a)]. \tag{52}$$

Also define the root-mean-square centering error relative to the exact optimal SDO along the same realized sequence by

$$\Delta^2_K \coloneqq \frac{1}{\eta^2 T}\sum_{k=0}^{T-1} \mathbb{E}_{s\sim d_k}[(c^K_k(s) - c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}_{k+1}}}(s))^2]. \tag{53}$$

Write the normalized logit-displacement range

$$\mathrm{rng}_k(s) \coloneqq \max_{a\in\mathcal{A}}[z^s_{\boldsymbol{\theta}_{k+1}}(a) - z^s_{\boldsymbol{\theta}_k}(a)] - \min_{a\in\mathcal{A}}[z^s_{\boldsymbol{\theta}_{k+1}}(a) - z^s_{\boldsymbol{\theta}_k}(a)] \tag{54}$$

Let $\varepsilon_{c^K,T}$ denote the normalized residual $\varepsilon_{c,T}$ evaluated at $c_k = c^K_k$. Then

$$0 \le \varepsilon^2_{c^K,T} - \varepsilon^2_T = \Delta^2_K \le \frac{1}{T}\sum_{k=0}^{T-1} \mathbb{E}_{s\sim d_k}\left[\left(1-\alpha^K_k(s)\right)^2\left(\frac{\mathrm{rng}_k(s)}{\eta}\right)^2\right]. \tag{55}$$

Equation (55) shows that the omitted probability mass affects the squared normalized residual through $\left(1-\alpha^K_k(s)\right)^2$. Thus, when the normalized logit-displacement range remains controlled, retaining most of the old-policy probability mass makes the additional Top-K residual small.

Corollary 1 (Average regret with Top-K approximation)**.** *Under the conditions of Theorem 1,*

$$\frac{1}{T}\sum_{k=0}^{T-1}\left(J(\pi^\star)-J(\pi_k)\right)\le \mathcal{C}_1 T^{-\frac{1}{2}}+\mathcal{C}_2\sqrt{\varepsilon_T^2+\Delta_K^2}, \tag{56}$$

where $\mathcal{C}_1 \coloneqq B_A\sqrt{2D_0}/(1-\gamma)$ *and* $\mathcal{C}_2 \coloneqq 2\sqrt{C_\mu}/(1-\gamma)$.

Therefore, the Top-K approximation leaves the $\mathcal{C}_1 T^{-1/2}$ optimization term unchanged and affects only the approximation term through Δ_K^2.

Proof. Fix k and s, and write $\Delta z_k^s(a) \coloneqq z^s_{\boldsymbol{\theta}_{k+1}}(a) - z^s_{\boldsymbol{\theta}_k}(a)$. When $\alpha_k^K(s) < 1$, define the mean displacement over the omitted actions by

$$c_k^{\overline{K}}(s) \coloneqq \sum_{a\notin\mathcal{T}^K_{k,s}} \frac{\pi_k(a|s)}{1-\alpha_k^K(s)}\Delta z_k^s(a). \tag{57}$$

The exact optimal SDO satisfies

$$c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}_{k+1}}}(s) = \alpha_k^K(s)c_k^K(s) + (1-\alpha_k^K(s))c_k^{\overline{K}}(s). \tag{58}$$

Consequently,

$$\left|c_k^K(s) - c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}_{k+1}}}(s)\right| \le (1-\alpha_k^K(s))\,\mathrm{rng}_k(s). \tag{59}$$

The same inequality holds trivially when $\alpha_k^K(s) = 1$. Squaring Equation (59), dividing by η^2, and averaging over states and iterations prove Equation (55).

Taking $c_k = c_k^K$ gives

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}_{(s,a)\sim\mu_k}\left[\left(\varepsilon^s_{k,c_k^K}(a)\right)^2\right] = \varepsilon_T^2+\Delta_K^2. \tag{60}$$

Applying Lemma 2 with $c_k = c_k^K$ and $\eta = \sqrt{2D_0/(B_A^2 T)}$ proves Equation (56). □

B Experimental Details

B.1 Details of the Two-State Illustrative Experiment

MDP specification. We use the two-state MDP of Kakade (2001) with states $\mathcal{S} = \{s_i, s_j\}$ and actions $\mathcal{A} = \{\text{stay}, \text{switch}\}$. The transition and reward functions are summarized in Table 3. The initial-state distribution is concentrated on s_i, with $\rho_0 = (1,0)^\top$, and the discount factor is $\gamma = 0.95$.

Table 3: Transition and reward functions of the two-state MDP.

State	Action	Next state	Reward
s_i	stay	s_i	1
s_i	switch	s_j	0
s_j	stay	s_j	2
s_j	switch	s_i	0

Policy parameterization and initialization. We use the shared logit map $g(\boldsymbol{\theta}) = \boldsymbol{z}_{\boldsymbol{\theta}}$ and order the action logits as $(\text{switch}, \text{stay})$. The switch logit is fixed to zero at both states: $\boldsymbol{z}_{\boldsymbol{\theta}}^{s_i} = \begin{bmatrix}0\\ \theta_1\end{bmatrix}, \boldsymbol{z}_{\boldsymbol{\theta}}^{s_j} = \begin{bmatrix}0\\ \theta_2+\beta\theta_1^2\end{bmatrix}, \beta = 1$. The stay logit therefore equals the stay-versus-switch log-odds, $z^s_{\boldsymbol{\theta}}(\text{stay}) = \log\frac{\pi_{\boldsymbol{\theta}}(\text{stay}|s)}{\pi_{\boldsymbol{\theta}}(\text{switch}|s)}$, and the action probabilities are $\pi_{\boldsymbol{\theta}}(\text{stay}|s) = \sigma(z^s_{\boldsymbol{\theta}}(\text{stay}))\,, \pi_{\boldsymbol{\theta}}(\text{switch}|s) = 1 -$

$\sigma(z^s_{\boldsymbol{\theta}}(\text{stay}))$, where $\sigma(u) = (1 + \exp(-u))^{-1}$. The Jacobian of the two stay logits with respect to the shared parameters is $\boldsymbol{J}_{\boldsymbol{\theta}} = \frac{\partial}{\partial \boldsymbol{\theta}^\top} \begin{bmatrix} z^{s_i}_{\boldsymbol{\theta}}(\text{stay}) \\ z^{s_j}_{\boldsymbol{\theta}}(\text{stay}) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 2\beta\theta_1 & 1 \end{bmatrix}$.

All methods start from the same stay probabilities, $\pi_{\boldsymbol{\theta}_0}(\text{stay}|s_i) = 0.75, \qquad \pi_{\boldsymbol{\theta}_0}(\text{stay}|s_j) = 0.25.$ The initial stay logits and parameters are therefore

$$\begin{bmatrix} z^{s_i}_{\boldsymbol{\theta}_0}(\text{stay}) \\ z^{s_j}_{\boldsymbol{\theta}_0}(\text{stay}) \end{bmatrix} = \begin{bmatrix} \log 3 \\ -\log 3 \end{bmatrix}, \qquad \boldsymbol{\theta}_0 = \begin{bmatrix} \log 3 \\ -\log 3 - \beta(\log 3)^2 \end{bmatrix}. \tag{61}$$

Exact policy evaluation. All policy quantities are computed analytically without trajectory sampling. For the current policy $\pi_{\boldsymbol{\theta}_k}$, we construct

$$[\boldsymbol{P}_{\pi_{\boldsymbol{\theta}_k}}]_{s,s'} = \sum_{a\in\mathcal{A}} \pi_{\boldsymbol{\theta}_k}(a|s)P(s'|s,a), \quad r_{\pi_{\boldsymbol{\theta}_k}}(s) = \sum_{a\in\mathcal{A}} \pi_{\boldsymbol{\theta}_k}(a|s)r(s,a). \tag{62}$$

The value function, action-value function, normalized discounted state-visitation distribution, and return are then evaluated as

$$V^{\pi_{\boldsymbol{\theta}_k}} = (\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi_{\boldsymbol{\theta}_k}})^{-1} r_{\pi_{\boldsymbol{\theta}_k}}, \quad Q^{\pi_{\boldsymbol{\theta}_k}}(s,a) = r(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a)V^{\pi_{\boldsymbol{\theta}_k}}(s'), \tag{63}$$

$$\left(d^{\pi_{\boldsymbol{\theta}_k}}_{\rho_0}\right)^\top = (1-\gamma)\rho_0^\top (\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi_{\boldsymbol{\theta}_k}})^{-1}, \qquad J(\pi_{\boldsymbol{\theta}_k}) = \rho_0^\top V^{\pi_{\boldsymbol{\theta}_k}}. \tag{64}$$

Thus, all reported learning curves are deterministic and do not require averaging over random seeds.

Common quantities used by the updates. Each method evaluates the following quantities under its own current policy. Define the action-value difference and state-wise Fisher weight as

$$\Delta Q^{\pi_{\boldsymbol{\theta}_k}}(s) = Q^{\pi_{\boldsymbol{\theta}_k}}(s, \text{stay}) - Q^{\pi_{\boldsymbol{\theta}_k}}(s, \text{switch}), \; w_{s,k} = d^{\pi_{\boldsymbol{\theta}_k}}_{\rho_0}(s)\pi_{\boldsymbol{\theta}_k}(\text{stay}|s)\pi_{\boldsymbol{\theta}_k}(\text{switch}|s). \tag{65}$$

We collect them into $\Delta\mathbf{Q}^{\pi_{\boldsymbol{\theta}_k}} = \begin{bmatrix} \Delta Q^{\pi_{\boldsymbol{\theta}_k}}(s_i) \\ \Delta Q^{\pi_{\boldsymbol{\theta}_k}}(s_j) \end{bmatrix}, \boldsymbol{W}_k = \text{diag}(w_{s_i,k}, w_{s_j,k})$. The parameter-space policy-gradient direction, with the global factor $1/(1-\gamma)$ absorbed into the learning rate, is

$$(1-\gamma)\, \nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_k} = \boldsymbol{J}^\top_{\boldsymbol{\theta}_k} \boldsymbol{W}_k \Delta\mathbf{Q}^{\pi_{\boldsymbol{\theta}_k}}. \tag{66}$$

Vanilla policy gradient. Vanilla PG updates the shared parameters using

$$\boldsymbol{\theta}^{\text{PG}}_{k+1} = \boldsymbol{\theta}^{\text{PG}}_k + \alpha_{\text{PG}} \boldsymbol{J}^\top_{\boldsymbol{\theta}^{\text{PG}}_k} \boldsymbol{W}_k \Delta\mathbf{Q}^{\pi_{\boldsymbol{\theta}^{\text{PG}}_k}}, \qquad \alpha_{\text{PG}} = 0.7. \tag{67}$$

Natural policy gradient. The exact parameter-space Fisher matrix is

$$\boldsymbol{F}_{\boldsymbol{\theta}_k} = \boldsymbol{J}^\top_{\boldsymbol{\theta}_k} \boldsymbol{W}_k \boldsymbol{J}_{\boldsymbol{\theta}_k}. \tag{68}$$

NPG uses the damped update

$$\boldsymbol{\theta}^{\text{NPG}}_{k+1} = \boldsymbol{\theta}^{\text{NPG}}_k + \eta \left(\boldsymbol{F}_{\boldsymbol{\theta}^{\text{NPG}}_k} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{J}^\top_{\boldsymbol{\theta}^{\text{NPG}}_k} \boldsymbol{W}_k \Delta\mathbf{Q}^{\pi_{\boldsymbol{\theta}^{\text{NPG}}_k}}, \tag{69}$$

where $\eta = 0.25$ and $\lambda = 10^{-10}$. Both the gradient and Fisher matrix are computed exactly.

Exact LOCAL. With the switch logits fixed to zero, the ideal state-wise PMD target is represented by the stay-logit vector

$$\boldsymbol{t}_k = \begin{bmatrix} z^{s_i}_{\boldsymbol{\theta}_k}(\text{stay}) \\ z^{s_j}_{\boldsymbol{\theta}_k}(\text{stay}) \end{bmatrix} + \eta \Delta\mathbf{Q}^{\pi_{\boldsymbol{\theta}_k}}, \qquad \eta = 0.25. \tag{70}$$

This parameterization can realize any pair of stay logits. Exact LOCAL therefore matches the target using

$$\theta^{\text{exact}}_{1,k+1} = t_k(s_i), \qquad \theta^{\text{exact}}_{2,k+1} = t_k(s_j) - \beta\left(\theta^{\text{exact}}_{1,k+1}\right)^2. \tag{71}$$

Thus, the exact curve is obtained from the analytic inverse rather than from running a large number of inner optimization steps.

Approximate LOCAL. For binary policies, profiling out the optimal SDO reduces the LOCAL objective to the weighted regression problem

$$\mathcal{L}_k(\boldsymbol{\theta}) = \frac{1}{2} \sum_{s \in \mathcal{S}} w_{s,k} \left(z^s_{\boldsymbol{\theta}}(\text{stay}) - t_k(s) \right)^2 . \tag{72}$$

Both $\boldsymbol{W}_k$ and $\boldsymbol{t}_k$ are fixed at their values under $\pi_{\boldsymbol{\theta}_k}$. Starting from $\boldsymbol{\theta}_k^{(0)} = \boldsymbol{\theta}_k$, each inner full-batch gradient-descent step is

$$\boldsymbol{\theta}_k^{(m+1)} = \boldsymbol{\theta}_k^{(m)} - \alpha_{\text{inner}} \boldsymbol{J}_{\boldsymbol{\theta}_k^{(m)}}^{\top} \boldsymbol{W}_k \begin{bmatrix} z^{s_i}_{\boldsymbol{\theta}_k^{(m)}}(\text{stay}) - t_k(s_i) \\ z^{s_j}_{\boldsymbol{\theta}_k^{(m)}}(\text{stay}) - t_k(s_j) \end{bmatrix} . \tag{73}$$

where α_{inner} is the learning rate of the inner loop optimization. After $M \in \{50, 200\}$ inner steps, the outer policy parameters are updated as $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k^{(M)}$.

B.2 SHARED ACTOR-CRITIC IMPLEMENTATION FOR ATARI AND PROCGEN

To examine the stability of LOCAL under different neural network architectures, the Atari experiments use the Nature CNN architecture of Mnih et al. (2015), whereas the Procgen experiments use the IMPALA-style ResNet architecture (Espeholt et al., 2018). The raw episodic return in Atari and Procgen is converted into the human-normalized score (Mnih et al., 2015, HNS) and the min–max normalized score (Cobbe et al., 2020). We report the interquartile mean (IQM), arithmetic mean, and median. The IQM averages the middle 50% of the environment-level scores.

Atari architecture. For the Atari experiments, all algorithms use the same CNN architecture. Each observation is converted to a grayscale image of size 84×84 and stacked over four consecutive frames, resulting in an input tensor of size $4 \times 84 \times 84$. The feature extractor consists of three convolutional layers: a 32-channel 8×8 convolution with stride 4, a 64-channel 4×4 convolution with stride 2, and a 64-channel 3×3 convolution with stride 1, with a ReLU activation after each layer. The resulting feature map is flattened to a 3136-dimensional vector and projected to a shared 512-dimensional representation through a linear layer followed by ReLU. No additional policy or value MLP is used. The shared representation is directly mapped to the action logits and the scalar value prediction through separate output heads.

Atari Human-Normalized Score. For Atari, the raw episodic return is converted into the human-normalized score (Mnih et al., 2015), $\text{HNS} = 100 \times \frac{R_{\text{agent}} - R_{\text{random}}}{R_{\text{human}} - R_{\text{random}}}$, where R_{random} and R_{human} are the reference scores of random and human policies.

Procgen architecture. For the Procgen experiments, all algorithms use the same IMPALA-style residual CNN architecture (Espeholt et al., 2018). Each observation is an RGB image of size 64×64 and is stacked over four consecutive frames, resulting in an input tensor of size $12 \times 64 \times 64$. The feature extractor consists of three convolutional stages with channel sizes 16, 32, and 32. Each stage begins with a 3×3 convolution with stride 1 and padding 1, followed by a 3×3 max-pooling layer with stride 2 and two residual blocks. Each residual block contains two 3×3 convolutions with stride 1 and padding 1, with ReLU activations and an identity skip connection. For a 64×64 input, the three stages produce a feature map of size $32 \times 7 \times 7$, which is flattened to a 1568-dimensional vector and projected to a shared 256-dimensional representation through a linear layer followed by ReLU. No additional policy or value MLP is used. The shared representation is directly mapped to the action logits and the scalar value prediction through separate output heads.

Procgen Min–Max Normalized Score. For Procgen, we convert the episodic return into the min–max normalized score: $R_{\text{norm}} = 100 \times \frac{R_{\text{agent}} - R_{\text{low}}}{R_{\text{high}} - R_{\text{low}}}$, where R_{low} and R_{high} are the per-environment reference constants from Cobbe et al. (2020). For both benchmarks, we report the interquartile mean (IQM), arithmetic mean, and median across environments.

Experiment Setup. For each algorithm and environment, we run 5 independent trials with different random seeds on Atari and Procgen, and report the mean performance together with the

Algorithm 1 Shared actor–critic implementation of LOCAL

Require: Initial shared-encoder parameters ω_0, actor-head parameters $\boldsymbol{\theta}_0$, and critic-head parameters ϕ_0; target-logit scale η; value coefficient c_v; entropy coefficient c_{ent}.

1: **for** $k = 0, 1, 2, \ldots$ **do**

2: Roll out $\pi_{\omega_k,\boldsymbol{\theta}_k}$ to collect $\mathcal{D}_k = \{(s_i, a_i, r_i, s_i')\}_{i=1}^{N}$.

3: For every sampled state s_i, store the old-policy logits $\boldsymbol{z}^{s_i}_{\omega_k,\boldsymbol{\theta}_k}$, action probabilities $\boldsymbol{\pi}^{s_i}_{\omega_k,\boldsymbol{\theta}_k}$, and value prediction $V_{\omega_k,\phi_k}(s_i)$.

4: Estimate advantages $\{\widehat{A}_i\}_{i=1}^{N}$ and value targets $\{\widehat{R}_i\}_{i=1}^{N}$ using the stored old-policy value predictions.

5: Initialize the parameters used for fitting as $(\omega, \boldsymbol{\theta}, \phi) \leftarrow (\omega_k, \boldsymbol{\theta}_k, \phi_k)$.

6: **for** each minibatch $\mathcal{B} \subseteq \mathcal{D}_k$ **do**

7: For every s_t in $\mathcal{B}$, compute the optimal shift

$$c^\star(s_t; \omega, \boldsymbol{\theta}, \omega_k, \boldsymbol{\theta}_k) = \sum_{b \in \mathcal{A}} \pi_{\omega_k,\boldsymbol{\theta}_k}(b|s_t) \left[z^{s_t}_{\omega,\boldsymbol{\theta}}(b) - z^{s_t}_{\omega_k,\boldsymbol{\theta}_k}(b) \right].$$

8: Compute the actor loss

$$\mathcal{L}_{\text{actor}} = \frac{1}{2|\mathcal{B}|} \sum_{t \in \mathcal{B}} \left[z^{s_t}_{\omega,\boldsymbol{\theta}}(a_t) - z^{s_t}_{\omega_k,\boldsymbol{\theta}_k}(a_t) - c^\star(s_t; \omega, \boldsymbol{\theta}, \omega_k, \boldsymbol{\theta}_k) - \eta \widehat{A}_t \right]^2.$$

9: Compute the critic loss

$$\mathcal{L}_V = \frac{1}{2|\mathcal{B}|} \sum_{t \in \mathcal{B}} \left[V_{\omega,\phi}(s_t) - \widehat{R}_t \right]^2.$$

10: Compute the minibatch policy entropy

$$\mathcal{H}_{\mathcal{B}} = -\frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \sum_{b \in \mathcal{A}} \pi_{\omega,\boldsymbol{\theta}}(b|s_t) \log \pi_{\omega,\boldsymbol{\theta}}(b|s_t).$$

11: Update $(\omega, \boldsymbol{\theta}, \phi)$ by descending $\mathcal{L}_{\text{actor}} + c_v \mathcal{L}_V - c_{\text{ent}} \mathcal{H}_{\mathcal{B}}$.

12: **end for**

13: Set $(\omega_{k+1}, \boldsymbol{\theta}_{k+1}, \phi_{k+1}) \leftarrow (\omega, \boldsymbol{\theta}, \phi)$.

14: **end for**

corresponding variability across seeds. All compared algorithms use the same environment preprocessing, neural-network architecture, rollout collection procedure, and evaluation protocol within each domain. The shared actor–critic implementation of LOCAL is summarized in Algorithm 1.

Hardware. Each Atari and Procgen run uses a single NVIDIA L40S GPU.

Table 4: Shared experimental settings for All algorithms in Atari and Procgen.

Hyperparameter	Atari	Procgen
Total environment steps	30,000,000	25,000,000
Number of parallel environments	8	32
Rollout length per environment	128	256
Samples collected per rollout	1,024	8,192
Discount factor γ	0.99	0.999
GAE parameter λ	0.95	0.95
Frame stack	4	4
Training levels	n/a	100
Evaluation levels	n/a	100

Table 5: LOCAL hyperparameters for Atari and Procgen.

Hyperparameter	Atari	Procgen
Batch size	256	256
Update epochs	1	1
Learning rate	2.5×10^{-4}	2.5×10^{-4}
Entropy coefficient	0.01	0.01
Value-loss coefficient	0.5	0.5
Maximum gradient norm	0.5	0.5
Optimizer	Adam	Adam
Target-logit scale η	1.0	1.0

Table 6: PPO hyperparameters for Atari and Procgen.

Hyperparameter	Atari	Procgen
Batch size	256	256
PPO epochs	4	3
Learning rate	2.5×10^{-4}	5×10^{-4}
Entropy coefficient	0.01	0.01
Value-loss coefficient	0.5	0.5
Maximum gradient norm	0.5	0.5
Optimizer	Adam	Adam
Clipping range	0.1	0.2

Table 7: ACKTR hyperparameters for Atari and Procgen.

Hyperparameter	Atari	Procgen
Batch size	1024	1024
Actor learning rate	7×10^{-4}	1×10^{-3}
Entropy coefficient	0.01	0.01
Value-loss coefficient	0.5	0.5
Maximum gradient norm	0.5	0.5
Optimizer	K-FAC	K-FAC
K-FAC momentum	0.9	0.9
K-FAC damping	0.003	0.01
K-FAC exponential moving average	0.95	0.95
Covariance update frequency	1	1
Inverse update frequency	20	20

Table 8: V-MPO hyperparameters for Atari and Procgen.

Hyperparameter	Atari	Procgen
Batch size	256	1024
V-MPO update epochs	4	4
Learning rate	1×10^{-4}	1×10^{-4}
Entropy coefficient	0	0
Value-loss coefficient	0.5	0.5
Maximum gradient norm	0.5	0.5
Optimizer	Adam	Adam
Top-k fraction	0.5	0.5
KL constraint for temperature ϵ_η	0.01	0.02
KL constraint for policy update ϵ_α	0.002	0.01
Initial temperature η	0.5	1.0
Initial KL multiplier α	0.5	5.0
Dual learning rate	3×10^{-5}	3×10^{-5}
Target-network update interval	10	1

Table 9: MDPO hyperparameters for Atari and Procgen.

Hyperparameter	Atari	Procgen
Batch size	256	1024
update epochs	4	4
Learning rate	1×10^{-4}	1×10^{-4}
Entropy coefficient	0.0	0.0
Value-loss coefficient	0.5	0.5
Maximum gradient norm	0.5	0.5
Optimizer	Adam	Adam

B.3 Implementation for LLM post-training

Our LLM experiments use the `verl` framework, the open-source implementation of HybridFlow (Sheng et al., 2024), and implement LOCAL within its established and empirically stable on-policy post-training pipelines rather than attempting to reproduce idealized condition of the theory. In each paired comparison, we retain the standard rollout, advantage-estimation, and minibatch-optimization designs of PPO with GAE for RLHF and of PPO with GAE or GRPO with group-relative advantages for RLVR, replacing only the actor objective (Schulman et al., 2017; Shao et al., 2024).

Algorithm 2 Top-K LOCAL

Require: Initial actor parameters $\boldsymbol{\theta}$; optional critic parameters ϕ; Top-K size K; step size η.

for iteration $k = 0, 1, 2, \ldots$ **do**
  Rollout π_k to collect trajectories $\{(s_i, a_i, r_i, s_i', \log \pi_k(a_i|s_i))\}_{i=1}^N$.
  Estimate advantages $\{\widehat{A}_i\}$. ▷ Use GRPO or GAE
  if a critic is used **then**
    Estimate value targets $\{\widehat{R}_i\}$.
  end if
  For each sampled state-action pair (s_i, a_i), store the old-policy Top-K support:

$$\mathcal{T}_i^K = \mathrm{TopK}(\boldsymbol{\pi}_k^{s_i}, K)\,.$$

  for each actor minibatch $\mathcal{B}$ **do**
    Compute the Top-K approximated logit residual

$$\delta_t^K = z_{\boldsymbol{\theta}}^{s_t}(a_t) - z_{\boldsymbol{\theta}_k}^{s_t}(a_t) - \sum_{b \in \mathcal{T}_t^K} \frac{\pi_k(b|s_t)}{\sum_{j \in \mathcal{T}_t^K} \pi_k(j|s_t)} \left[z_{\boldsymbol{\theta}}^{s_t}(b) - z_{\boldsymbol{\theta}_k}^{s_t}(b) \right] - \eta \widehat{A}_t.$$

    Compute the actor loss:

$$\mathcal{L}_\pi = \frac{1}{2|\mathcal{B}|} \sum_{t \in \mathcal{B}} \left(\delta_t^K\right)^2.$$

    Update actor parameters $\boldsymbol{\theta}$ by descending $\mathcal{L}_\pi$.
  end for
  if a critic is used **then**
    Update critic parameters ϕ.
  end if
end for

Experiment setup for RLVR. For the RLVR experiments, we use four instruction-tuned models: Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct (Yang et al., 2024a), as well as Llama-3.2-3B-Instruct and Llama-3.1-8B-Instruct (Team, 2024). We evaluate the post-trained models on MATH-500 (Lightman et al., 2024), a hard subset of MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), GSM-P1 and GSM-P2 (Mirzadeh et al., 2025), GPQA (Rein et al., 2023), AMC 2023 (Yang et al., 2024b), MMLU-Pro (Wang et al., 2024), and HotpotQA (Yang et al., 2018).

Hardware. All RLHF and RLVR experiments use NVIDIA H200 GPUs. Each run uses one GPU for Qwen2.5-3B-Instruct and Llama-3.2-3B-Instruct, two GPUs for Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct.

Table 10: Large language model post-training configurations for RLHF and RLVR.

Configuration	RLHF	RLVR
Training data	`full_hh_rlhf` (Bai et al., 2022)	6,000 MATH (Hendrycks et al., 2021) + 1,500 GSM8K (Cobbe et al., 2021)
Reward model	`Skywork/Skywork-Reward-V2-Llama-3.2-3B`	-
Maximum prompt length	128	2,048
Maximum response length	128	2,048
Global prompt batch size	32	64
Prompt minibatch size	8	16
PPO epochs	1	1
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$)	AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$)
Actor learning rate	1×10^{-6}	1×10^{-6}
Critic learning rate	1×10^{-5}	1×10^{-5} (PPO Adv.)
Weight decay	0	0
Discount factor γ	1.0	1.0
GAE parameter λ	1.0	1.0 (PPO Adv.)
Random seed	42	42
Loss aggregation	`token-mean`	`token-mean`
Temperature	1.0	1.0
top-p	1.0	1.0
top-k	-1	-1
Group Size	1	8 (GRPO Adv.) / 1 (PPO Adv.)
Actor KL loss coefficient	0.0	0.0
PPO clip range (PPO / GRPO)	0.2	0.2
LOCAL η	1.0	5.0 (GRPO Adv.) / 1.0 (PPO Adv.)
LOCAL K	64	64

C ADDITIONAL RESULTS

C.1 ATARI GAMES

Figure 4 and Figure 5 presents the evaluation reward throughout training for 57 Atari environments. The solid curves show the mean performance over 5 random seeds, while the shaded regions indicate the standard deviation across seeds. Table 11 presents HNS for each environment, sorted by LOCAL.

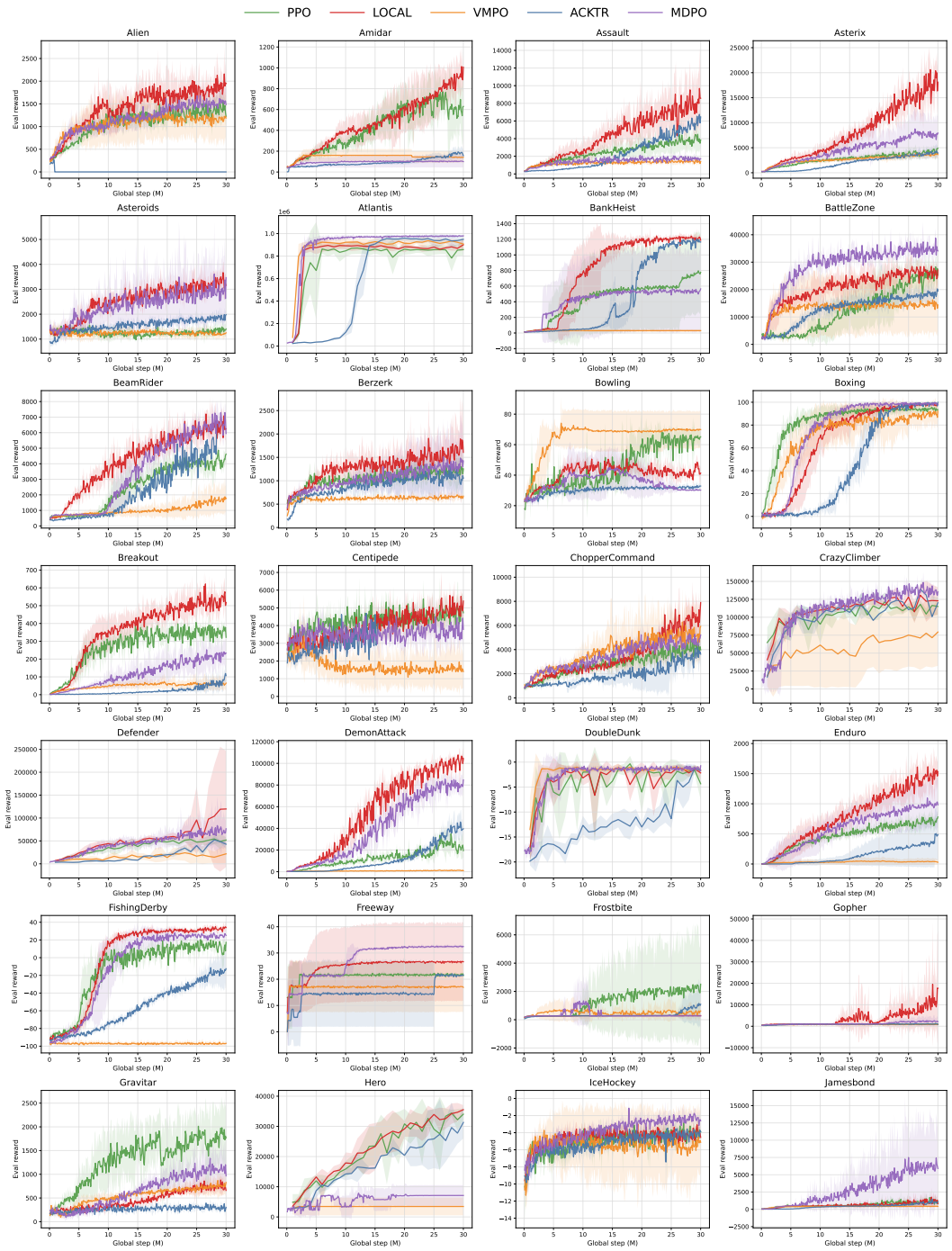


Figure 4: Evaluation reward vs. Environment steps of 57 Atari games (Part 1).

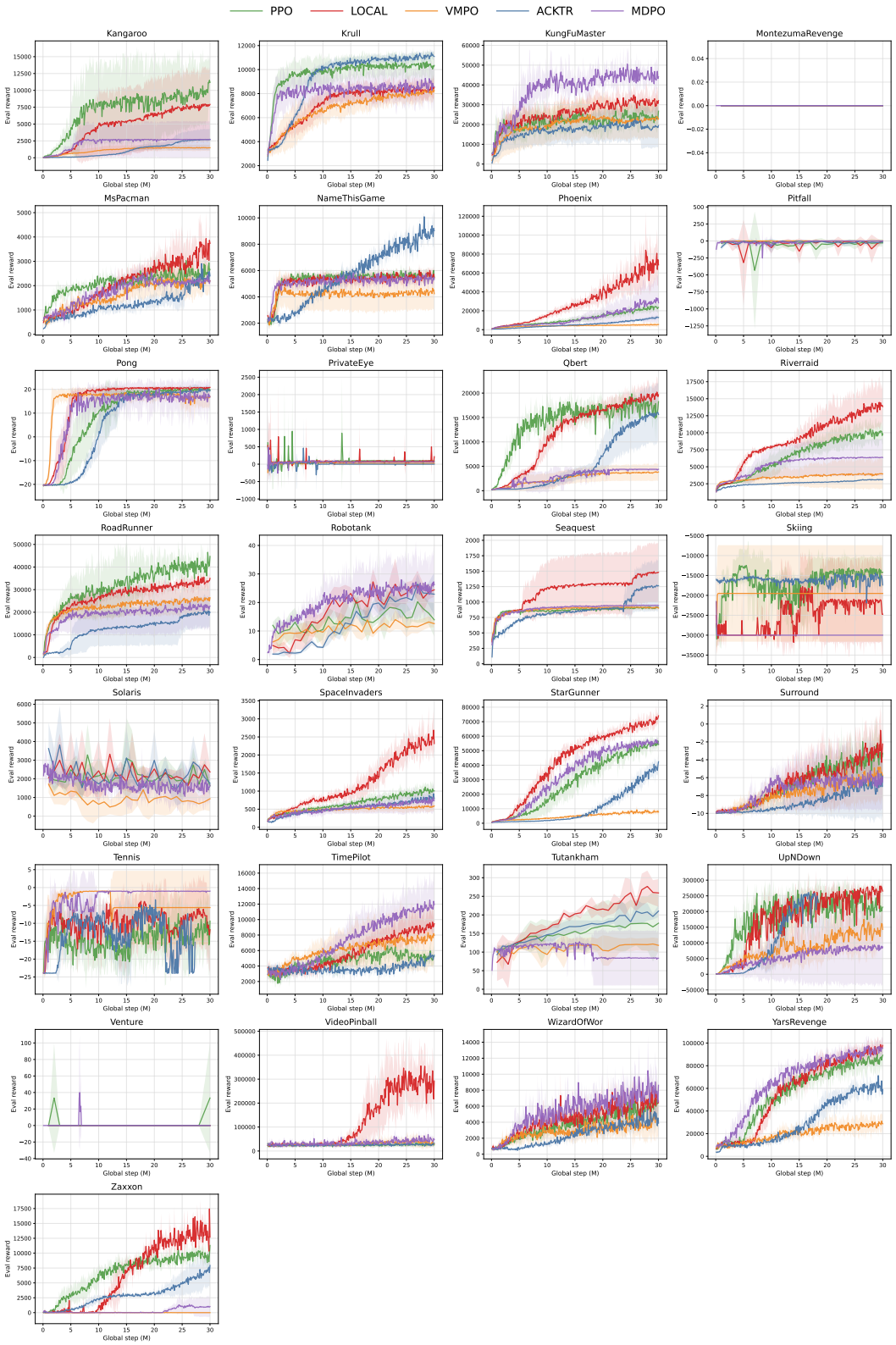


Figure 5: Evaluation reward vs. Environment steps of 57 Atari games (Part 2).

Table 11: Per-environment human-normalized scores (Human score: 100), sorted by LOCAL. Each entry reports the mean, with the sample standard deviation across distinct seeds shown as a superscript.

Environment	LOCAL	PPO	ACKTR	V-MPO	MDPO
VideoPinball	$\mathbf{19560.80}^{\pm 8594.37}$	$645.07^{\pm 766.07}$	$1169.67^{\pm 367.95}$	$942.02^{\pm 468.19}$	$2202.94^{\pm 1003.00}$
DemonAttack	$\mathbf{5736.70}^{\pm 184.43}$	$1058.23^{\pm 398.99}$	$2167.81^{\pm 648.73}$	$45.91^{\pm 27.12}$	$4648.66^{\pm 478.21}$
Atlantis	$5485.91^{\pm 322.58}$	$5219.97^{\pm 316.59}$	$5783.59^{\pm 230.87}$	$5529.59^{\pm 321.74}$	$\mathbf{5979.30}^{\pm 34.55}$
Breakout	$\mathbf{1857.29}^{\pm 426.46}$	$1112.04^{\pm 189.58}$	$365.71^{\pm 118.88}$	$227.00^{\pm 96.68}$	$803.94^{\pm 168.08}$
Assault	$\mathbf{1606.22}^{\pm 846.55}$	$695.26^{\pm 166.97}$	$826.06^{\pm 672.19}$	$228.23^{\pm 56.82}$	$289.15^{\pm 25.06}$
UpNDown	$1591.72^{\pm 1357.00}$	$\mathbf{1913.65}^{\pm 431.62}$	$811.43^{\pm 1100.71}$	$1275.78^{\pm 443.77}$	$774.75^{\pm 1094.18}$
Phoenix	$\mathbf{1047.34}^{\pm 361.46}$	$352.55^{\pm 40.29}$	$175.76^{\pm 27.38}$	$71.85^{\pm 31.84}$	$426.58^{\pm 313.73}$
Boxing	$\mathbf{817.50}^{\pm 8.74}$	$778.61^{\pm 18.57}$	$367.00^{\pm 411.90}$	$767.33^{\pm 63.89}$	$816.11^{\pm 16.67}$
Gopher	$\mathbf{810.52}^{\pm 1251.24}$	$33.21^{\pm 5.33}$	$29.90^{\pm 5.56}$	$28.44^{\pm 21.52}$	$86.49^{\pm 85.19}$
StarGunner	$\mathbf{765.76}^{\pm 66.08}$	$558.72^{\pm 43.54}$	$435.18^{\pm 62.59}$	$73.35^{\pm 20.71}$	$565.54^{\pm 19.79}$
DoubleDunk	$746.97^{\pm 62.47}$	$648.48^{\pm 145.33}$	$769.70^{\pm 17.50}$	$769.70^{\pm 17.50}$	$\mathbf{812.12}^{\pm 10.50}$
Defender	$\mathbf{739.18}^{\pm 802.27}$	$310.03^{\pm 40.64}$	$170.82^{\pm 148.90}$	$119.46^{\pm 126.62}$	$465.84^{\pm 300.19}$
Krull	$647.04^{\pm 55.49}$	$819.19^{\pm 54.43}$	$\mathbf{889.64}^{\pm 31.04}$	$619.40^{\pm 54.56}$	$650.52^{\pm 131.99}$
CrazyClimber	$447.70^{\pm 75.87}$	$416.26^{\pm 48.05}$	$257.37^{\pm 195.48}$	$273.98^{\pm 188.19}$	$\mathbf{498.01}^{\pm 31.37}$
RoadRunner	$446.55^{\pm 10.70}$	$\mathbf{570.52}^{\pm 21.80}$	$254.04^{\pm 80.32}$	$336.99^{\pm 84.52}$	$290.10^{\pm 138.01}$
TimePilot	$355.29^{\pm 119.01}$	$101.37^{\pm 67.00}$	$94.63^{\pm 43.13}$	$264.51^{\pm 216.84}$	$\mathbf{503.97}^{\pm 172.10}$
Jamesbond	$285.24^{\pm 109.45}$	$281.96^{\pm 46.83}$	$399.93^{\pm 143.54}$	$144.27^{\pm 58.51}$	$\mathbf{2144.87}^{\pm 2482.48}$
Kangaroo	$251.63^{\pm 144.78}$	$\mathbf{373.16}^{\pm 107.55}$	$88.90^{\pm 36.83}$	$47.94^{\pm 9.87}$	$89.44^{\pm 90.52}$
FishingDerby	$\mathbf{237.26}^{\pm 8.59}$	$205.66^{\pm 6.96}$	$149.43^{\pm 40.93}$	$-9.62^{\pm 4.79}$	$220.57^{\pm 0.38}$
Robotank	$229.04^{\pm 24.68}$	$120.79^{\pm 61.18}$	$213.57^{\pm 78.98}$	$107.04^{\pm 12.97}$	$\mathbf{247.42}^{\pm 110.35}$
Asterix	$\mathbf{215.72}^{\pm 53.27}$	$56.42^{\pm 6.91}$	$38.01^{\pm 20.26}$	$41.21^{\pm 10.50}$	$93.13^{\pm 41.46}$
YarsRevenge	$\mathbf{184.29}^{\pm 14.31}$	$167.47^{\pm 17.54}$	$110.79^{\pm 10.61}$	$49.75^{\pm 16.14}$	$179.32^{\pm 17.37}$
Enduro	$\mathbf{178.19}^{\pm 21.07}$	$90.75^{\pm 16.99}$	$42.53^{\pm 49.04}$	$3.39^{\pm 4.85}$	$114.06^{\pm 48.55}$
BankHeist	$\mathbf{158.94}^{\pm 6.34}$	$103.40^{\pm 68.57}$	$80.06^{\pm 62.44}$	$2.19^{\pm 2.31}$	$74.59^{\pm 74.93}$
Tutankham	$\mathbf{158.57}^{\pm 23.77}$	$106.50^{\pm 49.26}$	$121.17^{\pm 14.87}$	$68.35^{\pm 12.16}$	$46.22^{\pm 46.81}$
Qbert	$\mathbf{146.58}^{\pm 21.35}$	$136.08^{\pm 10.50}$	$116.19^{\pm 28.73}$	$27.67^{\pm 6.88}$	$31.75^{\pm 2.45}$
WizardOfWor	$144.44^{\pm 44.44}$	$140.27^{\pm 38.61}$	$79.16^{\pm 26.14}$	$86.19^{\pm 51.22}$	$\mathbf{192.14}^{\pm 14.92}$
SpaceInvaders	$\mathbf{142.74}^{\pm 12.05}$	$55.75^{\pm 8.41}$	$41.58^{\pm 6.66}$	$30.65^{\pm 8.41}$	$38.58^{\pm 20.68}$
KungFuMaster	$141.53^{\pm 18.95}$	$118.27^{\pm 18.30}$	$85.53^{\pm 37.41}$	$100.89^{\pm 32.26}$	$\mathbf{195.03}^{\pm 22.46}$
Zaxxon	$\mathbf{138.58}^{\pm 9.28}$	$123.27^{\pm 34.89}$	$86.48^{\pm 32.30}$	$-0.36^{\pm 0.00}$	$10.80^{\pm 18.39}$
Pong	$\mathbf{117.73}^{\pm 0.34}$	$114.31^{\pm 3.21}$	$101.31^{\pm 26.49}$	$106.73^{\pm 10.42}$	$106.14^{\pm 18.32}$
Hero	$\mathbf{115.80}^{\pm 5.09}$	$110.77^{\pm 6.44}$	$77.14^{\pm 44.20}$	$8.15^{\pm 9.86}$	$20.40^{\pm 11.51}$
ChopperCommand	$\mathbf{107.79}^{\pm 38.73}$	$48.18^{\pm 11.07}$	$40.94^{\pm 11.92}$	$77.80^{\pm 41.96}$	$66.94^{\pm 28.68}$
Freeway	$90.14^{\pm 50.39}$	$72.30^{\pm 1.76}$	$54.73^{\pm 36.52}$	$57.09^{\pm 31.93}$	$\mathbf{109.68}^{\pm 1.86}$
Riverraid	$\mathbf{79.23}^{\pm 18.53}$	$53.97^{\pm 10.12}$	$10.42^{\pm 1.54}$	$16.55^{\pm 13.70}$	$31.80^{\pm 17.27}$
Tennis	$77.55^{\pm 71.78}$	$93.03^{\pm 50.30}$	$35.48^{\pm 35.71}$	$117.42^{\pm 66.36}$	$\mathbf{147.10}^{\pm 0.00}$
BattleZone	$71.04^{\pm 14.37}$	$65.72^{\pm 17.44}$	$49.14^{\pm 7.52}$	$34.50^{\pm 31.36}$	$\mathbf{91.13}^{\pm 4.51}$
Berzerk	$\mathbf{59.16}^{\pm 14.28}$	$45.09^{\pm 10.41}$	$25.92^{\pm 13.32}$	$21.81^{\pm 3.96}$	$51.86^{\pm 33.40}$
Amidar	$\mathbf{58.40}^{\pm 8.83}$	$36.38^{\pm 16.20}$	$8.84^{\pm 3.54}$	$7.92^{\pm 3.33}$	$5.51^{\pm 3.22}$
IceHockey	$54.96^{\pm 4.08}$	$60.12^{\pm 2.81}$	$58.26^{\pm 6.56}$	$50.21^{\pm 30.14}$	$\mathbf{71.07}^{\pm 0.83}$
NameThisGame	$54.48^{\pm 9.80}$	$55.47^{\pm 6.43}$	$\mathbf{118.97}^{\pm 16.62}$	$33.84^{\pm 21.95}$	$52.61^{\pm 4.91}$
Surround	$\mathbf{47.03}^{\pm 22.77}$	$40.85^{\pm 14.86}$	$25.21^{\pm 28.59}$	$25.58^{\pm 18.85}$	$24.44^{\pm 22.48}$
MsPacman	$\mathbf{42.49}^{\pm 14.93}$	$35.48^{\pm 7.75}$	$18.71^{\pm 12.25}$	$28.39^{\pm 8.24}$	$28.57^{\pm 2.33}$
BeamRider	$33.58^{\pm 9.79}$	$22.84^{\pm 8.74}$	$9.63^{\pm 17.84}$	$8.61^{\pm 5.08}$	$\mathbf{33.70}^{\pm 6.32}$
Centipede	$\mathbf{33.19}^{\pm 7.88}$	$26.52^{\pm 10.96}$	$12.05^{\pm 7.35}$	$-6.93^{\pm 9.89}$	$17.07^{\pm 10.40}$
Alien	$\mathbf{24.93}^{\pm 4.45}$	$14.45^{\pm 3.27}$	$-3.30^{\pm 0.00}$	$13.93^{\pm 7.30}$	$18.19^{\pm 5.24}$
Gravitar	$15.32^{\pm 2.39}$	$\mathbf{50.14}^{\pm 19.74}$	$4.07^{\pm 1.27}$	$20.59^{\pm 10.22}$	$24.97^{\pm 7.16}$
Bowling	$13.24^{\pm 2.80}$	$30.56^{\pm 2.10}$	$7.09^{\pm 0.87}$	$\mathbf{33.98}^{\pm 8.81}$	$5.16^{\pm 0.25}$
Solaris	$\mathbf{10.21}^{\pm 3.48}$	$3.49^{\pm 4.00}$	$4.72^{\pm 5.80}$	$-2.73^{\pm 5.98}$	$3.88^{\pm 8.73}$
Asteroids	$5.22^{\pm 0.81}$	$1.48^{\pm 0.61}$	$2.70^{\pm 0.31}$	$1.14^{\pm 0.44}$	$\mathbf{5.89}^{\pm 4.12}$
Frostbite	$5.21^{\pm 0.55}$	$\mathbf{57.29}^{\pm 103.48}$	$24.01^{\pm 37.24}$	$12.98^{\pm 16.37}$	$4.70^{\pm 0.57}$
Pitfall	$\mathbf{3.43}^{\pm 0.00}$	$3.00^{\pm 0.85}$	$3.18^{\pm 0.50}$	$\mathbf{3.43}^{\pm 0.00}$	$\mathbf{3.43}^{\pm 0.00}$
Seaquest	$\mathbf{3.36}^{\pm 1.16}$	$2.01^{\pm 0.04}$	$2.62^{\pm 1.00}$	$1.98^{\pm 0.07}$	$2.08^{\pm 0.06}$
PrivateEye	$\mathbf{0.27}^{\pm 0.34}$	$0.10^{\pm 0.01}$	$-0.01^{\pm 0.05}$	$0.00^{\pm 0.07}$	$0.06^{\pm 0.08}$
MontezumaRevenge	$\mathbf{0.00}^{\pm 0.00}$	$\mathbf{0.00}^{\pm 0.00}$	$\mathbf{0.00}^{\pm 0.00}$	$\mathbf{0.00}^{\pm 0.00}$	$\mathbf{0.00}^{\pm 0.00}$
Venture	$0.00^{\pm 0.00}$	$\mathbf{2.81}^{\pm 5.61}$	$0.00^{\pm 0.00}$	$0.00^{\pm 0.00}$	$0.00^{\pm 0.00}$
Skiing	$-59.99^{\pm 82.23}$	$\mathbf{22.78}^{\pm 30.04}$	$3.79^{\pm 5.99}$	$-18.87^{\pm 94.95}$	$-101.10^{\pm 0.00}$

Algorithm	Atari HNS (55 environments)		
	IQM (± bootstrap SE)	Mean (± seed SD)	Median (± bootstrap SE)
LOCAL	**137.15**$^{\pm 4.00}$	**472.14**$^{\pm 18.74}$	**138.58**$^{\pm 6.81}$
V-MPO	45.15$^{\pm 2.07}$	217.40$^{\pm 11.32}$	34.50$^{\pm 3.64}$
ACKTR	73.56$^{\pm 3.65}$	285.78$^{\pm 15.10}$	77.14$^{\pm 7.24}$
PPO	104.19$^{\pm 2.67}$	321.89$^{\pm 19.86}$	93.03$^{\pm 9.33}$
MDPO	102.76$^{\pm 4.64}$	400.51$^{\pm 67.63}$	74.59$^{\pm 7.07}$

Table 12: Aggregate Atari human-normalized scores (Human score: 100) over 55 environments, excluding VideoPinball and Gopher.

C.2 PROCGEN

Figure 6 presents the evaluation reward throughout training for all evaluated Procgen environments. Each agent is trained on a fixed set of 100 levels and evaluated on the same 100 levels, so the curves report sample efficiency of methods. The solid curves show the mean performance over 5 random seeds, while the shaded regions indicate the standard deviation across seeds.

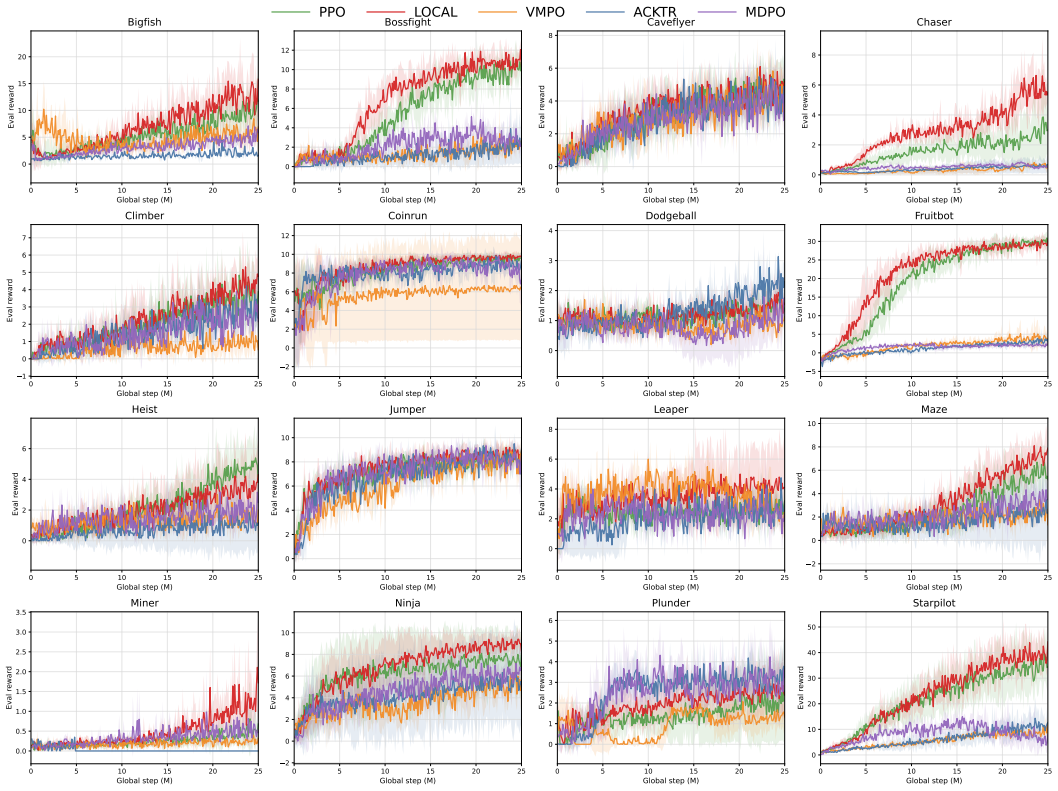


Figure 6: Evaluation reward on 100 levels vs. environment steps for all 16 Procgen games.

C.3 Additional Results for SDO ablation on Atari Games

We examine how the choice of SDO affects finite-sample target fitting in Atari. For both advantage and Q-value targets, we compare zero offset, uniform centering, and the optimized SDO $c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}}}(s)$. At the policy-space level, these SDOs represent the same exact state-wise softmax update. Under shared nonlinear function approximation, however, they define different fitting targets and can lead to different optimization behavior.

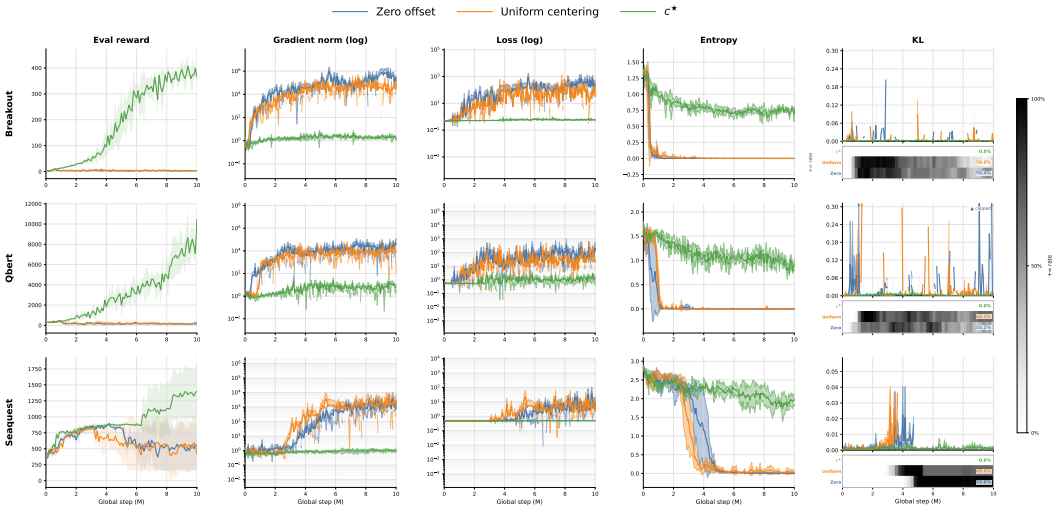


Figure 7: **SDO ablation with advantage-based targets on Atari.** Rows correspond to Breakout, Qbert, and Seaquest. Columns report evaluation reward, gradient norm and fitting loss on logarithmic scales, policy entropy, and KL divergence during training. In the KL column, triangular markers denote values clipped at the displayed upper limit, while the grayscale strips report the rate of $+\infty$ KL values; darker cells indicate higher rates, and the percentages on the right summarize the corresponding rates. The optimal SDO $c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}}}(s)$ produces more stable gradients and losses, avoids entropy collapse, and achieves the highest evaluation reward in all three environments.

Advantage-based targets. As shown in Figure 7, in Breakout and Qbert, zero offset and uniform centering are associated with rapidly increasing gradient norms and losses, early entropy collapse, and little reward improvement. In Seaquest, the three variants initially improve at similar rates, but the zero offset and uniform centering regress after their entropy decreases, whereas $c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}}}(s)$ retains nonzero entropy and continues improving. Moreover, the reported $+\infty$ KL rate is zero for $c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}}}(s)$ in all three environments, compared with rates between 49.6% and 63.2% for zero offset and uniform centering.

Q-value targets. As shown in Figure 8, the effect of the SDO is more pronounced for Q-value targets because the state-value component introduces an additional action-independent, state-dependent offset. Concretely, for a minibatch $\mathcal{B}$, the implementation minimizes the actor objective

$$\mathcal{L}^{Q}_{\text{actor}}(\boldsymbol{\theta}) = \frac{1}{2|\mathcal{B}|}\sum_{t\in\mathcal{B}}\left(\Delta z^{s_t}_{\boldsymbol{\theta}}(a_t) - c_{\boldsymbol{z}_{\boldsymbol{\theta}}}(s_t) - \eta\,\widetilde{Q}^{\lambda}_t\right)^2 - \frac{c_{\text{ent}}}{|\mathcal{B}|}\sum_{t\in\mathcal{B}}\mathcal{H}(\pi_{\boldsymbol{\theta}}(\cdot|s_t))\,, \tag{74}$$

where $\Delta z^s_{\boldsymbol{\theta}}(a) = z^s_{\boldsymbol{\theta}}(a) - z^s_{\boldsymbol{\theta}_{\text{old}}}(a)$. The sampled Q-value target is the standardized TD(λ) return, $\widehat{Q}^{\lambda}_t = V_{\boldsymbol{\theta}_{\text{old}}}(s_t) + \sum_{l\geq 0}(\gamma\lambda)^l\delta_{t+l}, \delta_t = r_t + \gamma V_{\boldsymbol{\theta}_{\text{old}}}(s_{t+1}) - V_{\boldsymbol{\theta}_{\text{old}}}(s_t), \widetilde{Q}^{\lambda}_t = \frac{\widehat{Q}^{\lambda}_t - \mu_{\mathcal{B}}}{\sigma_{\mathcal{B}}+10^{-8}}$, with the sum truncated at episode or rollout boundaries. The optimal SDO is computed exactly as $c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}}}(s) = \sum_{a\in\mathcal{A}}\pi_{\boldsymbol{\theta}_{\text{old}}}(a|s)\Delta z^s_{\boldsymbol{\theta}}(a)$.

For comparison, zero and uniform centering use $c(s) = 0$ and $c(s) = |\mathcal{A}|^{-1}\sum_a \Delta z^s_{\boldsymbol{\theta}}(a)$, respectively. In particular, using a Q-value target with zero offset recovers negative-entropy AMPO (Al-

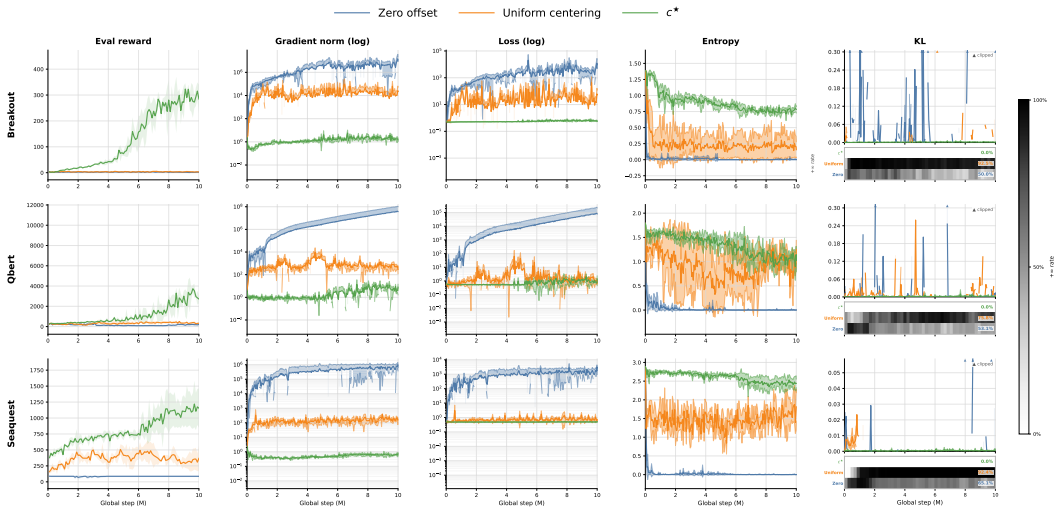


Figure 8: **SDO ablation with Q-value targets on Atari.** Rows correspond to Breakout, Qbert, and Seaquest, and the columns follow Figure 7. The Q-value target with zero offset is exactly the negative-entropy instance of AMPO. Compared with zero offset and uniform centering, the optimal SDO $c^\star_{\boldsymbol{z}_\theta}(s)$ consistently produces smaller gradient norms and fitting losses, maintains higher policy entropy, and achieves the highest evaluation reward. It also keeps the KL close to zero and has a zero reported $+\infty$ KL rate in all three environments.

fano et al., 2023). This variant exhibits large gradient norms and fitting losses, rapid entropy collapse, and low evaluation reward across the three environments. Uniform centering reduces the scale of the gradients and losses and preserves more entropy, but it remains less stable than $c^\star_{\boldsymbol{z}_\theta}(s)$ and obtains lower reward. These results show that optimizing the SDO removes fitting pressure associated only with the selected logit representative and improves the stability of the finite-sample update.

C.4 Additional Results for SDO Ablation in RLHF

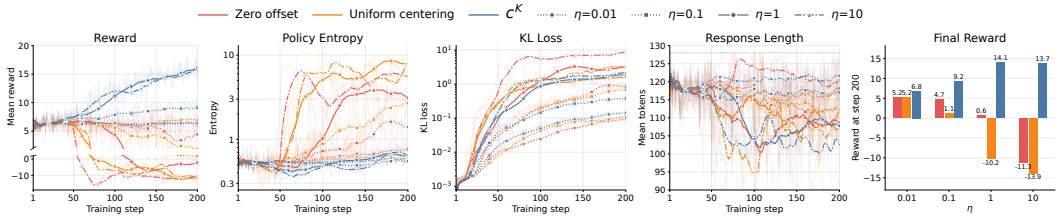


Figure 9: SDO and step-size ablation for RLHF training of Llama-3.2-3B-Instruct. From left to right, the panels show mean reward, policy entropy, KL loss, mean response length, and final reward at training step 200. Colors distinguish zero offset, uniform centering, and the Top-K SDO, while line styles and markers indicate $\eta \in \{0.01, 0.1, 1, 10\}$. The reward panel uses a split vertical axis to display both successful runs and collapsed runs. The horizontal dotted line in the response-length panel indicates the maximum response length of 128 tokens.

Figure 9 complements the RLVR ablation in the main text by testing the three SDO choices over a wide range of step sizes in RLHF. The Top-K SDO consistently achieves the highest final reward and remains stable as η increases, reaching rewards of 14.1 and 13.7 at $\eta = 1$ and $\eta = 10$, respectively. In contrast, zero offset and uniform centering become unstable at larger step sizes, accompanied by rapidly increasing entropy and KL loss and more variable response lengths.

	Checkpoint transition				Aggregate	Upper bound
K	100→200	200→300	300→400	400→468	$\widehat{R}_K$	Bound
16	0.10460	0.08648	0.10235	0.08453	0.09523	15.864
32	0.03015	0.02433	0.02969	0.02640	0.02776	4.003
64	0.01076	0.00955	0.01032	0.01039	0.01026	1.315
128	0.00476	0.00537	0.00453	0.00551	0.00501	0.576
256	0.00243	0.00346	0.00249	0.00325	0.00288	0.302

Table 13: Empirical relative centering error $\widehat{R}_K$ associated with Δ_K^2 in Equation (55), together with the corresponding normalized range-based upper bound. All entries are percentages.

This sensitivity is consistent with Proposition 2. The optimal SDO removes the action-independent logit component and preserves the state-wise Fisher geometry of the fitting residual. Zero offset and uniform centering introduce additional rank-one terms into the residual geometry, whose effect becomes more pronounced as the target displacement grows with η. The results therefore indicate that the optimal offset substantially widens the range of usable step sizes in RLHF.

C.5 Additional Results for Top-K Approximation Ablation

Equation (55) reduces the additional residual caused by Top-K to the centering error $(c_k^K(s) - c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}_{k+1}}}(s))^2/\eta^2$. To measure this term independently of the advantage scale, we evaluate the scale-free ratio

$$\widehat{R}_K := \frac{\sum_{k,s}(c_k^K(s) - c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}_{k+1}}}(s))^2}{\sum_{k,s}\mathbb{E}_{a\sim\pi_k(\cdot|s)}\left[(\delta z_k^s(a) - c^\star_{\boldsymbol{z}_{\boldsymbol{\theta}_{k+1}}}(s))^2\right]}, \qquad \delta z_k^s(a) := z^s_{\boldsymbol{\theta}_{k+1}}(a) - z^s_{\boldsymbol{\theta}_k}(a),$$

for which the factor η^{-2} cancels. We compute this quantity from full-vocabulary logits on 288,936 fixed MATH-500 states across four saved checkpoint transitions of the same trajectory. As shown in Table 13, $\widehat{R}_K$ decreases from 0.095% at $K = 16$ to 0.0029% at $K = 256$, supporting the prediction that the Top-K contribution Δ_K^2 is small relative to the intrinsic logit-update scale. The range-based upper bound is approximately 10^2 times larger, illustrating that it is conservative in practice.

C.6 Additional Results on Curvature Estimation Comparison with REBEL (Gao et al., 2024)

Curvature induced by sampled-pair regression. To isolate the effect of sampled pairing, we apply the paired-regression construction of REBEL (Gao et al., 2024) to token actions at a shared prefix, rather than to complete responses. Fix a state s and draw a, b independently from $\pi_{\boldsymbol{\theta}_k}(\cdot|s)$. With the overall loss scale matched to LOCAL, consider

$$\ell^s_{\text{pair}}(\boldsymbol{z}_{\boldsymbol{\theta}}; a, b) = \frac{1}{4}\left[\log\frac{\pi_{\boldsymbol{\theta}}(a|s)}{\pi_{\boldsymbol{\theta}_k}(a|s)} - \log\frac{\pi_{\boldsymbol{\theta}}(b|s)}{\pi_{\boldsymbol{\theta}_k}(b|s)} - \eta\big(A^{\pi_{\boldsymbol{\theta}_k}}(s,a) - A^{\pi_{\boldsymbol{\theta}_k}}(s,b)\big)\right]^2. \tag{75}$$

The softmax normalization terms cancel within each action pair:

$$\log\frac{\pi_{\boldsymbol{\theta}}(a|s)}{\pi_{\boldsymbol{\theta}_k}(a|s)} - \log\frac{\pi_{\boldsymbol{\theta}}(b|s)}{\pi_{\boldsymbol{\theta}_k}(b|s)} = z^s_{\boldsymbol{\theta}}(a) - z^s_{\boldsymbol{\theta}_k}(a) - z^s_{\boldsymbol{\theta}}(b) + z^s_{\boldsymbol{\theta}_k}(b). \tag{76}$$

Holding the old policy and advantage targets fixed, the logit-space Hessian is therefore

$$\nabla^2_{\boldsymbol{z}^s_{\boldsymbol{\theta}}}\ell^s_{\text{pair}} = \frac{1}{2}(\mathbf{e}_a - \mathbf{e}_b)(\mathbf{e}_a - \mathbf{e}_b)^\top, \tag{77}$$

where $\mathbf{e}_a$ denotes the coordinate vector corresponding to action a. This identity also holds when the advantages are replaced by fixed estimates. For m independent action pairs, the resulting curvature

estimator satisfies

$$\widehat{\boldsymbol{F}}_{s,\mathrm{pair}} = \frac{1}{2m}\sum_{i=1}^{m}(\mathbf{e}_{a_i} - \mathbf{e}_{b_i})(\mathbf{e}_{a_i} - \mathbf{e}_{b_i})^\top, \tag{78}$$

$$\mathbb{E}[\widehat{\boldsymbol{F}}_{s,\mathrm{pair}}] = \mathrm{diag}(\boldsymbol{\pi}^s_{\boldsymbol{\theta}_k}) - \boldsymbol{\pi}^s_{\boldsymbol{\theta}_k}{\boldsymbol{\pi}^s_{\boldsymbol{\theta}_k}}^\top = \boldsymbol{F}_s. \tag{79}$$

Thus, sampled-pair regression provides an unbiased estimate of the state-wise Fisher curvature, although a finite-sample realization generally differs from $\boldsymbol{F}_s$.

Bias introduced by Top-K centering. Top-K LOCAL generally introduces a curvature bias by approximating the old-policy center. Let $\mathcal{T}^K_{k,s}$ contain the K most probable actions under $\pi_{\boldsymbol{\theta}_k}(\cdot|s)$, and define $\alpha^K_k(s) = \sum_{a\in\mathcal{T}^K_{k,s}} \pi_{\boldsymbol{\theta}_k}(a|s), \widetilde{\pi}^K_{\boldsymbol{\theta}_k}(a|s) = \frac{\pi_{\boldsymbol{\theta}_k}(a|s)\mathbf{1}\{a\in\mathcal{T}^K_{k,s}\}}{\alpha^K_k(s)}$. Write $\widetilde{\boldsymbol{\pi}}^{s,K}_{\boldsymbol{\theta}_k}$ for the corresponding probability vector. For n independent action samples, the Top-K curvature estimator is

$$\widehat{\boldsymbol{F}}_{s,K} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{e}_{a_i} - \widetilde{\boldsymbol{\pi}}^{s,K}_{\boldsymbol{\theta}_k})(\mathbf{e}_{a_i} - \widetilde{\boldsymbol{\pi}}^{s,K}_{\boldsymbol{\theta}_k})^\top, \tag{80}$$

$$\mathbb{E}[\widehat{\boldsymbol{F}}_{s,K}] = \boldsymbol{F}_s + (\boldsymbol{\pi}^s_{\boldsymbol{\theta}_k} - \widetilde{\boldsymbol{\pi}}^{s,K}_{\boldsymbol{\theta}_k})(\boldsymbol{\pi}^s_{\boldsymbol{\theta}_k} - \widetilde{\boldsymbol{\pi}}^{s,K}_{\boldsymbol{\theta}_k})^\top. \tag{81}$$

Curvature estimation error. For independent samples $a_i \sim \pi_{\boldsymbol{\theta}_k}(\cdot \mid s)$, exact centering gives $\widehat{\boldsymbol{F}}_{s,\mathrm{LOCAL}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{e}_{a_i} - \boldsymbol{\pi}^s_{\boldsymbol{\theta}_k})(\mathbf{e}_{a_i} - \boldsymbol{\pi}^s_{\boldsymbol{\theta}_k})^\top$. For a fixed logit direction $\boldsymbol{v}$, independent of the action samples, the estimation errors at a matched budget of $n = 2m$ old-policy action samples satisfy

$$\begin{aligned}
\mathbb{E}\left[\left(\boldsymbol{v}^\top(\widehat{\boldsymbol{F}}_{s,\mathrm{pair}} - \boldsymbol{F}_s)\boldsymbol{v}\right)^2\right] &= \mathbb{E}\left[\left(\boldsymbol{v}^\top(\widehat{\boldsymbol{F}}_{s,\mathrm{LOCAL}} - \boldsymbol{F}_s)\boldsymbol{v}\right)^2\right] + \frac{2}{n}(\boldsymbol{v}^\top\boldsymbol{F}_s\boldsymbol{v})^2, \\
\mathbb{E}\left[\left(\boldsymbol{v}^\top(\widehat{\boldsymbol{F}}_{s,K} - \boldsymbol{F}_s)\boldsymbol{v}\right)^2\right] &= \mathrm{Var}\left(\boldsymbol{v}^\top\widehat{\boldsymbol{F}}_{s,K}\boldsymbol{v}\right) + \left[\boldsymbol{v}^\top(\boldsymbol{\pi}^s_{\boldsymbol{\theta}_k} - \widetilde{\boldsymbol{\pi}}^{s,K}_{\boldsymbol{\theta}_k})\right]^4.
\end{aligned} \tag{82}$$

Exact centering therefore has lower estimation variance than sampled pairing. Top-K centering can retain this advantage when its squared curvature bias is smaller than its variance reduction relative to sampled pairing. We examine this trade-off empirically below.

Experimental comparison. We test whether analytically computing the SDO improves curvature estimation over sampled pairing at a fixed action-sample budget, and whether Top-K centering retains this advantage. We use $S = 256$ fixed prefixes, taking two from each of 128 MATH500 problems. These prefixes are generated by the same frozen Llama-3.2-3B-Instruct checkpoint $\pi_{\boldsymbol{\theta}_0}$ used to evaluate curvature and sample actions. For each prefix s, we fix $D = 16$ random logit directions $\boldsymbol{v}_{s,j}$, remove their uniform component, and normalize them to unit Euclidean norm. We compute the exact directional curvature by summing over the full vocabulary:

$$\boldsymbol{v}_{s,j}^\top\boldsymbol{F}_s\boldsymbol{v}_{s,j} = \sum_{a\in\mathcal{A}}\pi_{\boldsymbol{\theta}_0}(a \mid s)\left(v_{s,j}(a) - \sum_{b\in\mathcal{A}}\pi_{\boldsymbol{\theta}_0}(b \mid s)v_{s,j}(b)\right)^2, \tag{83}$$

where $\boldsymbol{F}_s = \mathrm{diag}(\boldsymbol{\pi}^s_{\boldsymbol{\theta}_0}) - \boldsymbol{\pi}^s_{\boldsymbol{\theta}_0}{\boldsymbol{\pi}^s_{\boldsymbol{\theta}_0}}^\top$.

Exact-center LOCAL and Top-K LOCAL use $n \in \{4, 16, 32, 128, 256\}$ actions sampled independently with replacement from the full distribution $\pi_{\boldsymbol{\theta}_0}(\cdot \mid s)$. We consider two paired baselines: $n/2$ pairs using the same total of n actions, and n pairs using twice as many actions. Each prefix and budget combination is resampled $R = 1000$ times, with action samples shared among methods at matched budgets. For each method, let $\widehat{\boldsymbol{F}}^{(r)}_s$ denote its curvature estimate in repetition r. We report the NMSE

$$\mathrm{NMSE} = \frac{\sum_{s=1}^{S}\sum_{j=1}^{D}\frac{1}{R}\sum_{r=1}^{R}\left(\boldsymbol{v}_{s,j}^\top\widehat{\boldsymbol{F}}^{(r)}_s\boldsymbol{v}_{s,j} - \boldsymbol{v}_{s,j}^\top\boldsymbol{F}_s\boldsymbol{v}_{s,j}\right)^2}{\sum_{s=1}^{S}\sum_{j=1}^{D}\left(\boldsymbol{v}_{s,j}^\top\boldsymbol{F}_s\boldsymbol{v}_{s,j}\right)^2}. \tag{84}$$

Method	$n = 4$	$n = 16$	$n = 32$	$n = 128$	$n = 256$
Exact-center LOCAL	0.451328	0.113124	0.056782	0.014187	0.007093
Top-16 LOCAL	0.454303	0.113839	0.057139	0.014283	0.007145
Top-32 LOCAL	0.452469	0.113392	0.056916	0.014221	0.007111
Top-64 LOCAL	0.451859	0.113248	0.056844	0.014203	0.007101
Pairwise (n samples)	0.953183	0.239578	0.119826	0.029901	0.014969
Pairwise ($2n$ samples)	0.477137	0.119826	0.060111	0.014969	0.007476

Table 14: Normalized directional curvature MSE over 16 fixed random directions per prefix. Lower is better. All LOCAL variants use n sampled token actions per prefix. The paired estimators use $n/2$ pairs at the matched budget and n pairs at the doubled budget.

Results. At matched action-sample budgets, sampled pairing has approximately 2.1 times the NMSE of exact-center LOCAL (Table 14). Even when pairing uses twice as many actions, exact centering retains 5.1%–5.6% lower NMSE in these measurements. All tested Top-K variants remain within 0.8% of exact centering and outperform both paired baselines, indicating that the approximation preserves most of the benefit of analytic SDO computation in this setting. This comparison isolates local curvature estimation error between LOCAL and token-level paired regression; it does not assess the training performance of the original response-level REBEL algorithm.

C.7 REINFORCEMENT LEARNING WITH HUMAN FEEDBACK

C.7.1 ADDITIONAL RLHF DIAGNOSTICS

Figure 10 reports the per-model training dynamics of reward, entropy, KL loss, and response length. Across all four models, LOCAL maintains substantially higher entropy and generally achieves higher reward than PPO, while the KL loss and response length remain comparable.

C.7.2 REPRESENTATIVE RLHF GENERATIONS

We present representative RLHF generations where PPO and LOCAL use the same prompt and generation seed. We select positive-improvement cases near the median reward gain rather than the largest observed gains. The reward of each response is shown next to the corresponding method.

Llama-3.1-8B.

Human: What is the time difference between New York and Salt Lake City?

Assistant: 1 hour and 1 minute.

Human: Is Salt Lake City Central Time or Mountain Time?

PPO *Reward: 6.66*

Salt Lake City observes Mountain Time (MST). New York observes Eastern Time (EST). Therefore, Salt Lake City is 2 hours behind New York. Would you like to know the time difference as a straight hour difference or including the 1 minute difference I mentioned earlier? Clarification please!

LOCAL *Reward: 11.00*

Salt Lake City is Mountain Time (MT). It is not Central Time. During standard time, Salt Lake City is MT and New York is ET. When Daylight Saving Time (DST) is in effect, Salt Lake City is MDT and New York is EDT. The time difference with New York is 2 hours when DST is in effect. Note: Not all states observe DST, so this may not always apply—please verify DST status for specific locations.

Qwen2.5-7B.

Human: Does the container that a drink is in change the taste? I always think soda tastes better when its in a can.



Figure 10: Per-model RLHF training diagnostics for PPO and LOCAL. Each row corresponds to one model, and the columns show raw reward, entropy, KL loss, and response length.

PPO *Reward: 17.75*

The container can slightly affect taste due to factors like material flavor interaction, gas retention, and temperature. Cans often maintain carbonation and cooler temperatures better than bottles, which can enhance soda flavor. However, this effect is usually subtle. For most drinks, the primary factors affecting taste are ingredients, production methods, and storage conditions. To scientifically determine if container type changes taste, controlled sensory tests would be needed.

LOCAL *Reward: 21.12*

Whether it changes taste is debated among scientists & sommeliers. Factors influencing perception include container material, temperature, oxygen exposure & carbonation level rather than merely being in a can. Some studies suggest cans might preserve carbonation better. Personal preference also plays a big role. To test this, try blind tasting sessions controlling variables like temperature and storage.

C.8 Reinforcement Learning with Verifiable Rewards

C.8.1 Additional RLVR diagnostics

Figure 11 and Figure 12 reports the per-model training dynamics of training reward, validation accuracy, entropy, KL loss, and response length.

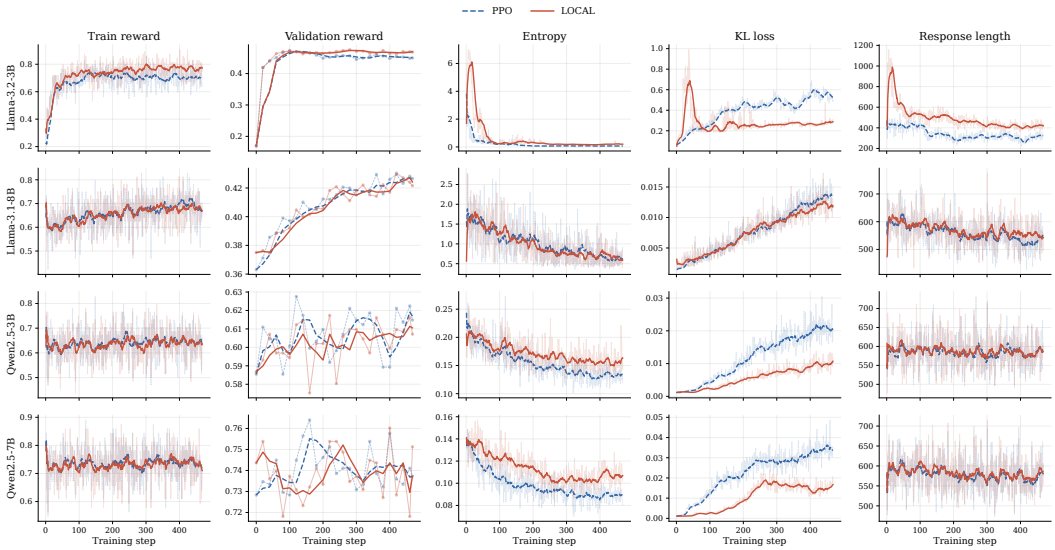


Figure 11: Per-model RLVR training diagnostics for PPO and LOCAL. Each row corresponds to one model, and the columns show training reward, validation accuracy, entropy, KL loss, and response length.

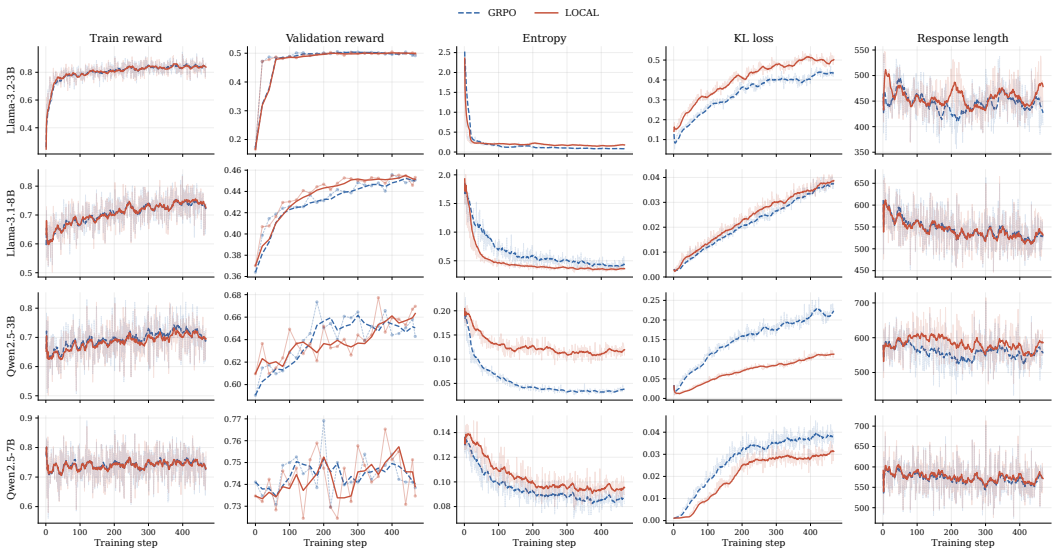


Figure 12: Per-model RLVR training diagnostics for GRPO and LOCAL. Each row corresponds to one model, and the columns show training reward, validation accuracy, entropy, KL loss, and response length.

C.8.2 REPRESENTATIVE RLVR GENERATIONS

Llama-3.1-8B (GRPO advantages).

Human: Pat is to select six cookies from a tray containing only chocolate chip, oatmeal, and peanut butter cookies. There are at least six of each of these three kinds of cookies on the tray. How many different assortments of six cookies can be selected? (Note that cookies of the same type are not distinguishable.) Let us think step by step and output the final answer within `\boxed{}`.

GRPO *Reward: 0*

Step 1: First, we need to understand the problem and the nature of selecting the cookies. Pat is selecting six cookies from a tray that contains only chocolate chip, oatmeal, and peanut butter cookies. There are at least six of each kind of cookie on the tray.
Step 2: Since the problem is asking for different assortments of six cookies, we need to determine all the possible combinations that can be made from these three types of cookies.
Step 3: To break this down, we'll consider the possible numbers of each type of cookie Pat can select. We'll use the variable x to represent the number of cookies of one type, y for another type, and z for the third type, and their sums will add up to 6 (since Pat is selecting six cookies in total).
Step 4: We need to consider all integer solutions to the equation x + y + z = 6, where x, y, and z are non-negative integers. However, since we know there are at least six of each type of cookie, we should focus on the combinations of (x, y, z) that satisfy this condition.
Step 5: We can approach this by listing out all the possible combinations that satisfy the conditions. Since the cookies of the same type are not distinguishable, the order does not matter.
Step 6: The possible combinations can be (6,0,0), (0,6,0), (0,0,6), (5,1,0), (5,0,1), (1,5,0), (0,5,1), (1,0,5), (0,1,5), (4,2,0), (4,0,2), (2,4,0), (0,4,2), (2,0,4), (0,2,4), (4,1,1), (1,4,1), (1,1,4), (3,3,0), (3,0,3), (0,3,3), (3,2,1), (3,1,2), (2,3,1), (1,3,2), (2,1,3), (1,2,3).
Step 7: Counting the number of combinations, we find that there are 21 possible ways to select the cookies.
The final answer is: $\boxed{21}$

LOCAL *Reward: 1*

Step 1: First, we need to understand the problem. Pat has to select six cookies from a tray that contains only chocolate chip, oatmeal, and peanut butter cookies. Each type of cookie is indistinguishable from the others within its kind. **Step 2:** The problem asks for the number of different assortments of six cookies that Pat can select. To solve this, we'll use a stars-and-bars approach to represent the selection of cookies. **Step 3:** Since we want to select 6 cookies out of 18 total cookies (as we have 6 of each kind and there are three kinds), we can represent the selection by placing 6 stars and 2 bars in a line, which separates the three types of cookies. **Step 4:** This gives us a total of 8 positions to fill: 6 for the stars (representing the cookies) and 2 for the bars (representing the separation between the types of cookies). **Step 5:** We need to calculate the number of ways to arrange 6 stars and 2 bars in a line, as this corresponds to the number of different selections of cookies. **Step 6:** To find the number of arrangements, we can use combinations. The number of ways to arrange the stars and bars is given by the combination formula $\binom{n}{k} = \frac{n!}{k!(n-k)!}$, where n is the total number of objects (in this case, 8) and k is the number of stars (also 6). **Step 7:** Plugging these values into the combination formula, we get $\binom{8}{6} = \frac{8!}{6!(8-6)!}$. **Step 8:** Calculating this, we get $\binom{8}{6} = \frac{8\times7}{2\times1}$. **Step 9:** Simplifying further, we get $\binom{8}{6} = 28$. **Step 10:** Therefore, there are 28 different assortments of six cookies that Pat can select.
The final answer is: $\boxed{28}$