EVPO: Explained Variance Policy Optimization for Adaptive Critic Utilization in LLM Post-Training

Anonymous ACL submission

Abstract

Reinforcement learning (RL) for LLM post-training faces a fundamental design choice: whether to use a learned critic as a baseline. Classical theory favors critic-based methods such as PPO for variance reduction, yet critic-free alternatives like GRPO are widely adopted for simplicity and competitive performance. We show that in sparse-reward settings, a learned critic can inject estimation noise exceeding the state signal it captures, increasing rather than reducing advantage variance. Casting baseline selection as a Kalman filtering problem, we unify PPO and GRPO as two extremes of the Kalman gain, proving that *explained variance* (EV), computable from a single training batch, identifies the exact boundary: positive EV indicates the critic reduces variance; zero or negative EV signals it inflates variance. We propose **E**xplained **V**ariance **P**olicy **O**ptimization (**EVPO**)[1], which monitors batch-level EV at each step and adaptively switches between critic-based and batch-mean advantage estimation, provably achieving no greater variance than the better of the two. Across four tasks spanning classical control, agentic interaction, and mathematical reasoning, EVPO consistently outperforms both PPO and GRPO, regardless of which fixed baseline is stronger. Analysis confirms that the adaptive gating tracks critic maturation over training and that the theoretically derived zero threshold is empirically optimal.

1 Introduction

Reinforcement learning (RL) has become a central paradigm for LLM post-training, driving advances in mathematical reasoning (Guo et al., 2025; Singh et al., 2025; Qwen Team, 2026; Yang et al., 2025), instruction following (Zhou et al., 2023; Ding et al., 2026; Dou et al., 2026), and interactive decision-making (Shridhar et al., 2021; Wang et al., 2025; Chen et al., 2025). A prominent instantiation is *reinforcement learning with verifiable rewards* (RLVR) (Guo et al., 2025), where a sparse reward is assigned only at the terminal step via an automatic verifier. A fundamental algorithmic choice in this setting is whether to employ a *critic* (a learned value function) as a baseline for policy optimization.

Classical RL theory holds that a critic reduces policy-gradient variance by providing a state-dependent baseline (Sutton et al., 1999; Schulman et al., 2018), forming the foundation of actor-critic methods including PPO (Schulman et al., 2017). In practice, however, critic-free methods such as GRPO (Shao et al., 2024; Guo et al., 2025) have gained widespread adoption, as they are cheaper and can match or even surpass PPO (Hu et al., 2025). Rather than treating the critic as a binary design choice, we ask:

In sparse-reward LLM post-training, when does a learned critic help and when does it hurt?

To answer this, we identify three progressive sub-questions. $\mathcal{Q}_1$: *Is the boundary observable?* Critic quality fluctuates during training, and we need to measure its impact on policy optimization in a controlled setting. $\mathcal{Q}_2$: *Where exactly is the boundary?* In sparse-reward LLM generation, the critic may inject more estimation noise than the state signal it captures. We need a precise characterization of when this happens. $\mathcal{Q}_3$: *Can this boundary be exploited online?* If the helpful-vs-harmful regime can be detected in real time, we can build an algorithm that adapts rather than committing to a fixed choice.

We adopt *explained variance* (EV), the fraction of return variance accounted for by the critic, as a real-time proxy for critic usefulness. Empirically ($\mathcal{Q}_1$), EV remains negative during the early stage of

[1] Our code is available at https://anonymous.4open.science/r/EVPO-9785.

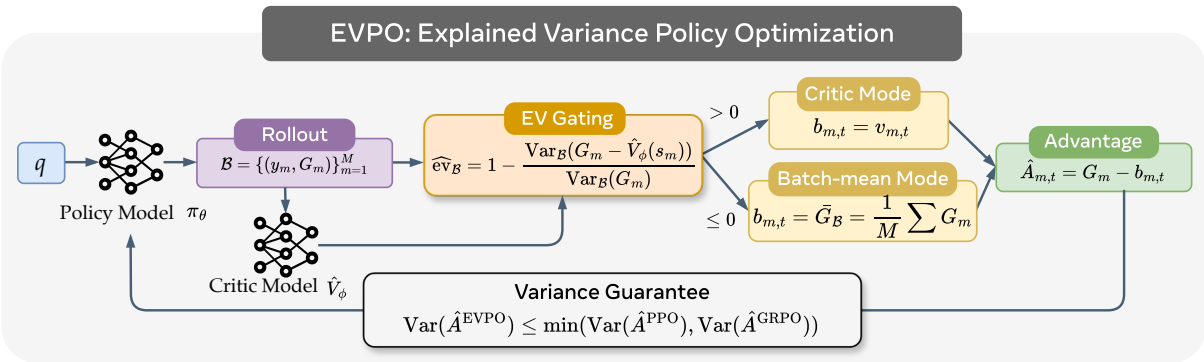


Figure 1: Overview of EVPO. The EV gating module selects between the critic baseline (EV > 0) and the batch-mean baseline (EV ≤ 0), guaranteeing advantage variance no greater than the minimum of PPO and GRPO.

training, and controlled interventions (suppressing the critic or injecting noise) confirm that performance degrades once EV falls to zero or below, suggesting that its sign reliably indicates whether the critic helps or hurts (Section 2.2). To pinpoint the boundary ($\mathcal{Q}_2$), we cast baseline selection as a Kalman filtering problem (Kalman, 1960) in which the critic and the batch-mean return are two competing estimators of the true state value, with PPO and GRPO corresponding to the two extremes of the Kalman gain (Section 2.3). We prove that the sign of EV, computable from a single batch, is sufficient to determine which extreme is preferable (Theorem 1), placing the boundary at EV $= 0$.

Exploiting this boundary online ($\mathcal{Q}_3$), we propose **Explained Variance Policy Optimization (EVPO)**, which implements this adaptive hard gating at each training step. EVPO maintains a standard PPO-style critic alongside the policy, and at each iteration computes batch-level EV as a scalar indicator. When EV is positive, the critic captures meaningful state-dependent signal, and EVPO uses its predictions as the baseline (the PPO mode); when EV falls to zero or below, EVPO discards the critic and falls back to the batch-mean return (the GRPO mode). We prove that this switching rule yields advantage-estimator variance no greater than the better of PPO and GRPO at every step, so EVPO never underperforms either fixed choice.

We evaluate EVPO on four tasks spanning classical control, interactive decision-making, and mathematical reasoning. Across all tasks, EVPO achieves the best final performance and maintains its advantage throughout training, regardless of whether PPO or GRPO is the stronger fixed baseline on a given task. Further analysis confirms that the adaptive gating tracks critic maturation over training and that the theoretically motivated zero threshold is empirically optimal.

Our main contributions are:

1. **Indicator and theory.** We unify PPO and GRPO as two extremes of a Kalman baseline-selection spectrum and prove that the sign of explained variance is the exact boundary separating the variance-reducing from the variance-inflating critic regime (Theorem 1).

2. **Algorithm.** We propose EVPO, a lightweight method that switches between critic-based and critic-free advantage estimation via batch-level EV, with a provable guarantee of no greater variance than the better of PPO and GRPO at every step.

3. **Empirical validation.** EVPO consistently outperforms fixed PPO and GRPO across four tasks spanning classical control, agentic interaction, and mathematical reasoning. Analysis experiments confirm that the zero threshold is empirically optimal and that the adaptive gating reflects critic maturation across training stages.

2 Preliminaries

This section establishes that the critic is not uniformly beneficial. We review PPO and GRPO as two baseline-selection extremes (Section 2.1), then show empirically that the sign of explained variance reliably predicts whether the critic helps or hurts (Section 2.2).

2.1 RL for LLM Post-Training

We model autoregressive text generation as a token-level Markov Decision Process (OpenAI et al.,

2024; Trung et al., 2024; Zheng et al., 2023). Given a prompt x, the policy π_θ generates a response $y = (y_1, \ldots, y_T)$ token by token, where the state at step t is $s_t = (x, y_{<t})$ and the action is $a_t = y_t \sim \pi_\theta(\cdot \mid s_t)$.

In the reinforcement learning with verifiable rewards (RLVR) setting (Guo et al., 2025; Wang et al., 2025), a scalar reward is assigned only at the terminal step: $r_T = \mathcal{R}(x, y)$, $r_t = 0$ for $t < T$. With discount factor $\gamma = 1$, the return from any state equals the terminal reward: $G_t = r_T \triangleq G$. The state-value function is $V^\pi(s) = \mathbb{E}_\pi[G \mid s_t = s]$, and the observed return G is a stochastic realization whose variance arises solely from the randomness of future actions under π.

PPO and GRPO. Both PPO (Schulman et al., 2017) and GRPO (Guo et al., 2025; Shao et al., 2024) optimize a clipped surrogate objective but differ in how they estimate advantages. Under RLVR conditions (γ=λ=1, terminal-only reward), the PPO advantage reduces to (see Appendix A.2 for the full GAE derivation):

$$\hat{A}_t^{\text{PPO}} = G - \hat{V}_\phi(s_t), \tag{1}$$

where $\hat{V}_\phi$ is a learned critic providing a *state-dependent* baseline. GRPO eliminates the critic and instead computes a *response-level* advantage from batch statistics. Concretely, for each prompt x, a batch of K responses $\{y^{(i)}\}_{i=1}^K$ is sampled from the current policy, yielding returns $\{G_i\}_{i=1}^K$ with batch mean $\bar{G}_\mathcal{B} = \frac{1}{K}\sum_{i=1}^K G_i$ and standard deviation $\sigma(G_\mathcal{B})$. The GRPO advantage is then:

$$\hat{A}_i^{\text{GRPO}} = \frac{G_i - \bar{G}_\mathcal{B}}{\sigma(G_\mathcal{B})}, \tag{2}$$

where $\bar{G}_\mathcal{B}$ and $\sigma(G_\mathcal{B})$ are the batch mean and standard deviation of returns, serving as a *constant* baseline shared across all tokens.

2.2 The Impact of Critic Quality on Policy Optimization

Returning to $\mathcal{Q}_1$ (is the boundary observable?), we quantify critic quality via the *explained variance* (EV) metric, a standard indicator from regression analysis (Schulman, 2016):

$$\text{ev} = 1 - \frac{\text{Var}(G - \hat{V}_\phi(s))}{\text{Var}(G)}, \tag{3}$$

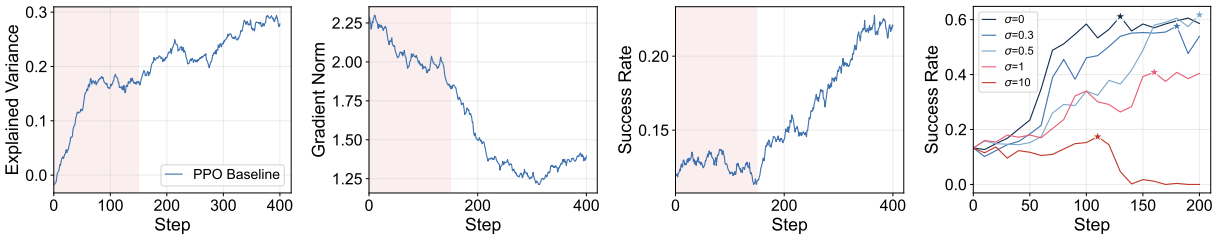
which measures the fraction of return variance accounted for by the critic. Intuitively, ev $>$ 0 indicates that the critic captures meaningful state-dependent signal, whereas ev $\leq$ 0 means the critic's estimation noise exceeds the signal it provides—its predictions are worse than a constant baseline.

Observation 1: A noisy critic hurts the actor. We train a PPO policy on Sokoban and track the batch-level EV throughout training (see Appendix D.1 for details). As shown in Figure 2a, EV remains consistently low during the first ∼150 training steps, indicating that the critic's predictions are less accurate than a simple constant baseline during this phase. This poor critic quality has two visible consequences: gradient norms are notably elevated (Figure 2b), reflecting high-variance policy updates driven by noisy advantage estimates, and the training success rate remains low and unstable (Figure 2c), showing that the actor struggles to learn under unreliable critic guidance.

To further confirm the causal link between critic quality and actor performance, we conduct a controlled experiment: injecting Gaussian noise ($\sigma_\delta = 0.3$; the mean absolute state value is ≈ 0.5) into a *converged* critic's outputs. As shown in Figure 2d, performance degrades monotonically as the noise level increases, with complete collapse at σ=10 where EV is driven far below zero. This confirms that **negative EV is a reliable indicator of harmful critic influence**, and that the relationship between critic quality and actor performance is not merely correlational but causal.

Observation 2: Reducing critic variance helps the actor. We conduct two complementary interventions on the same Sokoban and FrozenLake setups (see Appendix D.2 for details): (i) *Cold-start*: replacing the critic's output with the batch mean $\bar{G}_\mathcal{B}$ during the first 50/25 steps, and (ii) *Warm-up*: pre-training the critic alone for 200 steps before the actor begins updating. Both interventions yield faster convergence and more stable training curves (Figure 6), confirming that **suppressing critic variance during low-quality phases directly benefits the policy**.

These observations suggest that the choice between a learned critic and a batch-mean baseline should not be fixed a priori, but should adapt to the critic's current quality. In the following subsection, we formalize this intuition through a Kalman filtering lens.



(a) Explained Variance (b) Gradient Norm (c) Train Success (d) Noise Injection

Figure 2: Training dynamics of PPO on Sokoban. **(a)** Batch-level EV over training steps. **(b)** Policy gradient norms. **(c)** Training success rate. The shaded region in (a–c) marks the early phase where EV is low. **(d)** Effect of injecting Gaussian noise into a converged critic on FrozenLake.

2.3 A Kalman Filtering Perspective on Baseline Selection

To precisely locate the boundary ($\mathcal{Q}_2$), we observe that choosing a baseline is fundamentally a *state-estimation* problem: both the critic $\hat{V}_\phi(s)$ and the batch mean $\bar{G}_\mathcal{B}$ are noisy estimates of the true value $V^\pi(s)$, and the best baseline is the one with lower estimation error. This is exactly the setting addressed by the Kalman filter (Kalman, 1960; Pei et al., 2019), which optimally fuses two noisy estimates by weighting each inversely to its error variance.

Setup. The observed return decomposes as $G = V^\pi(s) + \epsilon$ with sampling noise $R \triangleq \mathbb{E}[\mathrm{Var}(G \mid s)]$. The two competing baselines have distinct error structures: (i) the *critic* $\hat{V}_\phi(s) = V^\pi(s) + \delta(s)$ with estimation error variance $P_A \triangleq \mathrm{Var}(\delta)$, and (ii) the *batch mean* $\bar{G}_\mathcal{B}$ whose deviation from any specific state value has variance $P_B \triangleq \mathrm{Var}_s(V^\pi(s))$. The minimum-variance linear combination of these two estimators assigns the critic a weight of $1 - K^*$ with the optimal gain $K^* = P_A/(P_A + P_B)$ (see Appendix A.1 for the full derivation).

Connection to explained variance. The EV metric (Eq. (3)) directly encodes the relative reliability of the two estimators. Substituting the variance decomposition into the definition of EV yields:

$$\mathrm{ev} = \frac{P_B - P_A}{P_B + R}. \tag{4}$$

Since the denominator $P_B + R$ is always positive, the *sign* of EV is determined solely by $P_B - P_A$. This means $\mathrm{ev} = 0$ **marks the exact boundary where** $P_A = P_B$, i.e., where the critic's estimation noise equals the state signal it captures. Above this boundary ($\mathrm{ev} > 0$), the critic outperforms the batch mean; below it ($\mathrm{ev} \leq 0$), critic noise dominates. The sampling noise R compresses the magnitude of EV but preserves its sign, keeping this boundary robust under high stochasticity.

3 Methodology

Building on the Kalman filtering framework of Section 2.3, we now formalize when a critic helps or hurts (Section 3.1), unify PPO and GRPO as two extremes of the Kalman gain (Section 3.2), and derive the EVPO algorithm (Section 3.3).

3.1 The Performance Collapse Boundary

From Eq. (4), the sign of EV is governed by $P_B - P_A$. We formalize this as:

Theorem 1 (Performance Collapse Boundary)**.** *Let $\hat{V}_\phi(s) = V^\pi(s) + \delta(s)$ with $\delta \perp \epsilon$, and let $P_A = \mathrm{Var}(\delta)$, $P_B = \mathrm{Var}_s(V^\pi(s))$, $K = \frac{P_A}{P_A + P_B}$. Then the following are equivalent:*

$$\mathrm{ev} \leq 0 \iff P_A \geq P_B \iff K \geq \tfrac{1}{2}. \tag{5}$$

In words, the critic inflates advantage variance whenever its estimation noise (P_A) exceeds the state signal it captures (P_B), and $\mathrm{ev} = 0$ is the exact tipping point. Since the sign of EV is computable from a single batch, this boundary is directly observable during training.

Computing EV from a training batch. Replacing the population variances in Eq. (3) with sample variances over a batch of M rollouts $\mathcal{B} = \{(y_m, G_m)\}_{m=1}^M$ from the same prompt gives the batch-level EV:

$$\widehat{\mathrm{ev}}_\mathcal{B} = 1 - \frac{\mathrm{Var}_\mathcal{B}(G_m - \hat{V}_\phi(s_m))}{\mathrm{Var}_\mathcal{B}(G_m)}, \tag{6}$$

computed from the critic values already obtained for advantage estimation, with no additional forward passes.

Algorithm 1 Explained Variance Policy Optimization (EVPO)

Require: Policy π_θ, critic $\hat{V}_\phi$, task set $\mathcal{D}$, group size M, number of tasks N

for each training iteration **do**
  Sample N tasks $\{x_j\}_{j=1}^{N} \sim \mathcal{D}$
  for $j = 1, \ldots, N$ **do**
    Rollout a batch with M trajectories: $\mathcal{B} = \{y_m\}_{m=1}^{M} \sim \pi_\theta(\cdot \mid x_j)$
    Obtain rewards $G_m = \mathcal{R}(x_j, y_m)$;
    compute critic values $v_{m,t} \leftarrow \hat{V}_\phi(s_{m,t})$
    $\widehat{\mathrm{ev}}_{\mathcal{B}} \leftarrow 1 - \frac{\mathrm{Var}_{\mathcal{B}}(G_m - v_m)}{\mathrm{Var}_{\mathcal{B}}(G_m)}$
    if $\widehat{\mathrm{ev}}_{\mathcal{B}} > 0$ **then**
      $b_{m,t} \leftarrow v_{m,t}$
    else
      $b_{m,t} \leftarrow \bar{G}_{\mathcal{B}} = \frac{1}{M}\sum_{m=1}^{M} G_m$
    end if
    $\hat{A}_{m,t} \leftarrow G_m - b_{m,t}$
  end for
  Aggregate gradients over N tasks;
  update π_θ and $\hat{V}_\phi$
end for

3.2 PPO and GRPO as Kalman Extremes

Under RLVR conditions (γ=λ=1, terminal-only reward), PPO and GRPO can be unified into the form $\hat{A}_t = G - b(s_t)$, differing only in the choice of baseline $b(s_t)$ (see Appendix A.2 for the detailed derivation). This difference maps directly onto the Kalman gain K:

- **PPO** (critic mode, K=0, full critic trust): $b(s_t) = \hat{V}_\phi(s_t)$. The baseline is entirely model-driven; the Kalman filter places zero weight on the evidence-driven estimator.
- **GRPO** (batch-mean mode, K=1, no critic trust): $b(s_t) = \bar{G}_{\mathcal{B}}$. (Since $1/\sigma(G_{\mathcal{B}})$ does not affect the gradient direction.) The baseline is entirely evidence-driven; the critic is discarded and the batch mean serves as a constant baseline for all tokens.

In other words, PPO and GRPO are not fundamentally different algorithms but rather two extreme operating points of the same Kalman baseline-selection spectrum. PPO implicitly assumes $P_A \ll P_B$ (the critic is always reliable), while GRPO implicitly assumes $P_A \gg P_B$ (the critic is never worth using). Neither assumption holds uniformly across training, motivating an adaptive approach.

3.3 EVPO: Adaptive Baseline Selection via EV Gating

Having established *where* the collapse boundary lies ($\mathcal{Q}_2$), we now turn to $\mathcal{Q}_3$: *can this boundary be exploited online?* The Kalman framework prescribes the optimal baseline as a continuous interpolation of the two estimators:

$$b(s) = (1-K)\,\hat{V}_\phi(s) + K\,\bar{G}_{\mathcal{B}}, \qquad K = \frac{P_A}{P_A + P_B} \in [0,1]. \tag{7}$$

However, computing K directly is infeasible: both $P_A = \mathrm{Var}(\hat{V}_\phi(s) - V^\pi(s))$ and $P_B = \mathrm{Var}_s(V^\pi(s))$ depend on the unknown true value $V^\pi(s)$, making the continuous K non-identifiable in practice. Two remedies are available. The first is a pragmatic fallback: replace the unsamplable $V^\pi(s)$ with the samplable return G, which yields an estimate systematically biased toward $\frac{1}{2}$. We call this approach *Continual* K (Cont. K; see Appendix A.1 for the derivation); in our experiments we find that such an overly moderate estimate fails to exploit the strengths of either PPO or GRPO.

The second remedy—adopted by EVPO—is to further reduce the domain of K from the continuous $[0,1]$ to the binary set $\{0,1\}$, corresponding to the two extremes: K=0 (PPO, full critic trust) and K=1 (GRPO, no critic). The decision boundary $\frac{P_A}{P_A+P_B} = \frac{1}{2}$ (i.e., $P_A = P_B$) corresponds exactly to $\mathrm{ev} = 0$ by Theorem 1. While the *value* of K is non-identifiable, its position relative to $\frac{1}{2}$ can be determined by the sign of $\widehat{\mathrm{ev}}_{\mathcal{B}}$ (Eq. (6)). This yields the **EVPO switching rule**:

$$b(s_t) = \begin{cases} \hat{V}_\phi(s_t), & \text{if } \widehat{\mathrm{ev}}_{\mathcal{B}} > 0, \\ & (K < \tfrac{1}{2}: \text{ trust critic}) \\ \bar{G}_{\mathcal{B}}, & \text{if } \widehat{\mathrm{ev}}_{\mathcal{B}} \le 0, \\ & (K \ge \tfrac{1}{2}: \text{ retreat to batch mean}) \end{cases} \tag{8}$$

where $\widehat{\mathrm{ev}}_{\mathcal{B}}$ is the batch-level EV defined in Eq. (6).

Variance guarantee. By construction, EVPO selects the lower-variance baseline at each step:

$$\mathrm{Var}(\hat{A}^{\mathrm{EVPO}}) = R + \min(P_A,\ P_B), \tag{9}$$

which is no greater than $\mathrm{Var}(\hat{A}^{\mathrm{PPO}}) = R + P_A$ or $\mathrm{Var}(\hat{A}^{\mathrm{GRPO}}) = R + P_B$ at every training step. Over the course of training, this per-step guarantee accumulates: EVPO avoids the critic's noisy cold-start phase ($P_A \gg P_B$) while fully exploiting a mature critic ($P_A \ll P_B$).

4 Experiments

Our experiments aim to validate that the per-step variance reduction of EVPO (Eq. 9) translates into consistent performance gains over fixed baselines (Section 4.2). We further analyze whether EV serves as a reliable indicator of the P_A-vs-P_B balance throughout training (Section 4.3) and whether the theoretically derived collapse boundary $\mathrm{ev} = 0$ (Theorem 1) coincides with the empirically optimal switching threshold (Section 4.4).

4.1 Experimental Setup

Evaluation. We evaluate on four tasks that span diverse agent interaction paradigms. **Sokoban** (Junghanns and Schaeffer, 2001) and **FrozenLake** (Dell'Aversana, 2021) are multi-turn grid-world games requiring sequential planning under sparse rewards, with FrozenLake additionally introducing environmental stochasticity. **WebShop** (Yao et al., 2022) is an interactive web-based agentic task involving multi-step instruction following and decision making. **MATH** targets competition-level mathematical reasoning trained and evaluated on problems in DAPO-Math-17k (Yu et al., 2025).

Baseline. We compare EVPO against the **Base LLM** without RL post-training, two critic-based methods (**PPO** (Schulman et al., 2017) and **StarPO-S** (Wang et al., 2025)), and two critic-free methods (**GRPO** (Shao et al., 2024) and **DAPO** (Yu et al., 2025)). For the Math task we use Qwen2.5-7B-Instruct (Qwen et al., 2025) as the base model; the other three tasks use Qwen2.5-3B-Instruct (Qwen et al., 2025). To ensure fair comparison, the hyperparameter recipe is tuned independently per task but kept consistent across all methods on the same task. Full configurations are provided in Appendix B.

4.2 Main Results

Finding 1: EVPO improves over both PPO and GRPO regardless of which fixed baseline is stronger on a given task.

The stronger fixed baseline varies across tasks: critic-based methods (PPO, StarPO-S) clearly outperform critic-free methods (GRPO, DAPO) on Sokoban, FrozenLake, and WebShop, while critic-free methods are more competitive on MATH. EVPO surpasses both families in every case (Table 1), confirming that it is not tied to a single favorable regime but adapts to the task-specific P_A-vs-P_B balance.

Table 1: Best validation success rate across four tasks. **Bold** indicates the best result per task. For each task, the success rate is calculated based on a total of 512 experimental results, obtained by sampling 16 times for each of the 32 queries.

Method	Sokoban	FrozenLake	WebShop	MATH
Base LLM	0.139	0.133	0.025	0.332
Critic-based methods				
PPO	0.314	0.611	0.252	0.373
StarPO-S	0.455	0.359	0.178	0.371
Critic-free methods				
GRPO	0.156	0.234	0.143	0.379
DAPO	0.162	0.207	0.127	0.385
Cont. K	0.244	0.490	0.242	0.371
EVPO(*ours*)	**0.604**	**0.684**	**0.303**	**0.416**

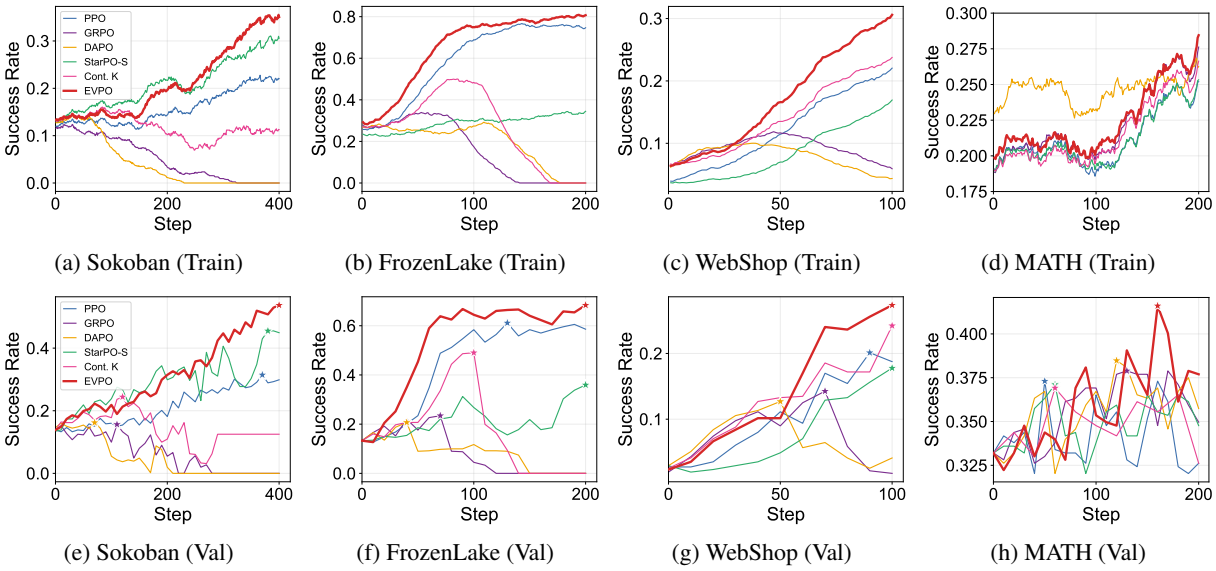
Finding 2: EVPO's advantage persists throughout training, not just at convergence.

As shown in Figure 3, EVPO leads or matches the strongest baseline at most training stages across all four tasks. This indicates that adaptive switching takes effect continuously—not by recovering from a poor start or catching up late, but by maintaining lower-variance advantage estimates throughout the optimization trajectory. The per-step variance guarantee (Eq. 9) predicts exactly this behavior: since EVPO selects the lower-variance baseline at every step, its cumulative benefit should manifest as a sustained advantage rather than a late-stage correction.

4.3 Gating Behavior Analysis

Finding 3: In most tasks, gating frequency decreases monotonically over training, reflecting critic maturation from $P_A > P_B$ to $P_A < P_B$.

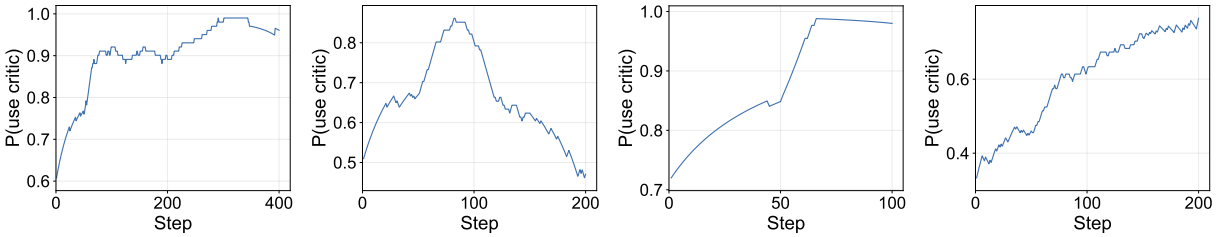
Figure 4 tracks the critic-selection probability over training. Across Sokoban, WebShop, and MATH, the batch-mean mode is triggered frequently in early training and progressively less as training continues, consistent with the expectation that P_A decreases as the critic improves. A notable exception is FrozenLake, where gating frequency reverses around step 80, coinciding with training stagnation (Figure 3(b)). This suggests that when the policy enters a difficult region, critic quality can temporarily regress, and the adaptive gating responds by retreating to the batch-mean baseline.



(a) Sokoban (Train) (b) FrozenLake (Train) (c) WebShop (Train) (d) MATH (Train)

(e) Sokoban (Val) (f) FrozenLake (Val) (g) WebShop (Val) (h) MATH (Val)

Figure 3: Training and validation curves across four tasks. **Top row:** training success rate (smoothed). **Bottom row:** validation success rate (unsmoothed; stars mark each method's peak). EVPO tracks or exceeds the best competing method throughout most of training.



(a) Sokoban (b) FrozenLake (c) WebShop (d) MATH

Figure 4: Gating trigger probability ($\widehat{\mathrm{ev}}_{\mathcal{B}} > 0$, i.e., selection of critic over batch mean) across training.

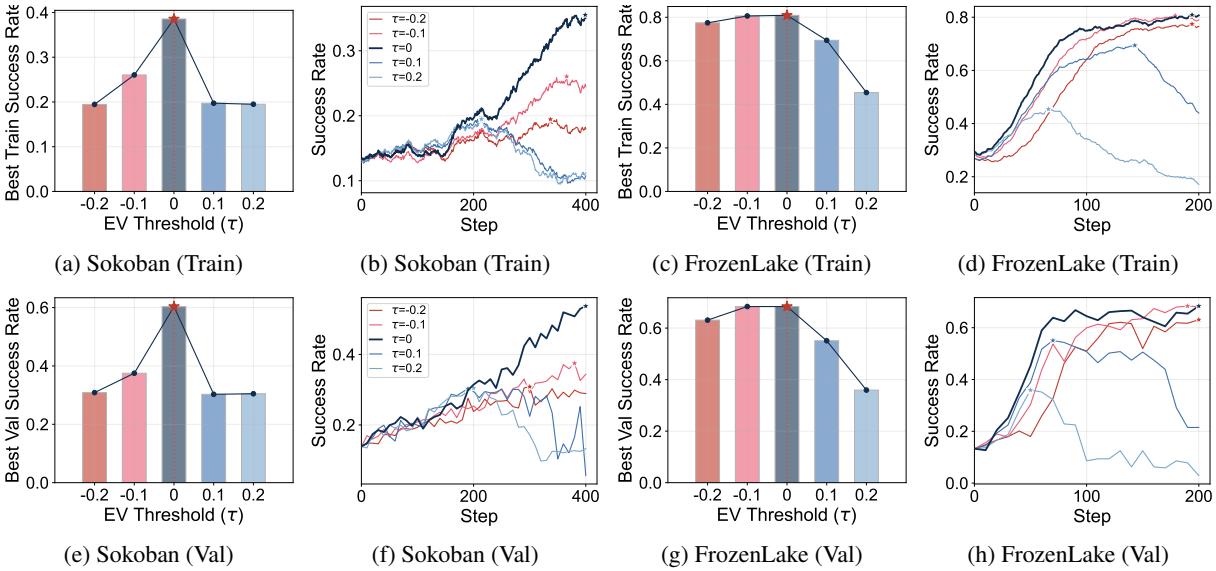
4.4 EV Threshold Sensitivity

Finding 4: Performance peaks at $\mathrm{ev} = 0$ and degrades in either direction, with positive and negative deviations inducing distinct failure modes.

We vary the switching threshold across $\{-0.2, -0.1, 0, 0.1, 0.2\}$ while keeping all other configurations fixed. As shown in Figure 5(a, c, e, g), the theoretically derived boundary $\mathrm{ev} = 0$ (Theorem 1) is not merely a theoretical convenience but the practically optimal operating point. The training landscape in Figure 5(b, d, f, h) reveals an asymmetry in the two deviation directions. A positive threshold under-utilizes the critic: training progresses normally in early stages but collapses later when policy refinement requires state-dependent advantage estimates that the batch mean cannot provide. A negative threshold over-utilizes the critic: the immature critic's noise slows early training, and in later stages, unreliable value predictions on certain state subsets continue to inject variance. These two failure modes reveal that, as we argued theoretically, the zero threshold is precisely the boundary separating the $P_A > P_B$ and $P_A < P_B$ regimes.

5 Related Work

Reinforcement Learning with Verifiable Rewards for LLMs. RL-based LLM post-training has evolved from actor-critic architectures such as PPO (Schulman et al., 2017), which rely on a learned value function for variance reduction via GAE (Schulman et al., 2018), toward critic-free alternatives that bypass the value network entirely. GRPO (Shao et al., 2024; Guo et al., 2025; Wang et al., 2026) replaces the learned baseline with intra-group reward normalization and has become the dominant paradigm; REINFORCE++ (Hu et al., 2025) further stabilizes this family via global batch normalization, and similar estimators like RLOO (Ahmadian et al., 2024) and DAPO (Yu et al., 2025) further demonstrate that a



(a) Sokoban (Train) (b) Sokoban (Train) (c) FrozenLake (Train) (d) FrozenLake (Train)

(e) Sokoban (Val) (f) Sokoban (Val) (g) FrozenLake (Val) (h) FrozenLake (Val)

Figure 5: Threshold sensitivity analysis. **Top row:** training metrics (smoothed); **bottom row:** validation metrics (unsmoothed). **(a, c, e, g)** Best success rate as a function of the EV switching threshold; stars mark τ=0. **(b, d, f, h)** Full curves under each threshold setting. Performance peaks at τ=0 on both tasks.

critic is often dispensable for single-turn, verifiable tasks.

Yet critic-free methods face two systematic challenges: group-relative advantages introduce statistical bias, systematically underestimating hard prompts and overestimating easy ones (Jain et al., 2025; Lin et al., 2026), and uniform trajectory-level credit assignment degrades severely in long-horizon settings lacking early termination (de Oliveira et al., 2025; Dong et al., 2025), compelling multi-turn frameworks such as RAGEN (Wang et al., 2025) to reintroduce the critic for stable training. All existing approaches resolve this tension between critic-free simplicity and critic-based credit assignment (Li et al., 2025) through a static, a priori architectural commitment that either always uses or always discards the critic, regardless of its current predictive quality.

Adaptive Baselines and Explained Variance. Reducing policy-gradient variance is a foundational RL challenge (Williams, 1992; Sutton et al., 1999). State-dependent baselines rely on the assumption that the value network is accurate; in LLM post-training, the vast state space and sparse rewards frequently violate the accuracy Prior work addresses this through two complementary strategies: redesigning the baseline structure itself (Anschel et al., 2017), or using critic quality as a signal to adapt hyperparameters such as λ in GAE (Mohammadpour et al., 2024) or to filter uninformative transitions (Flet-Berliac and Preux, 2019).

However, these methods treat critic quality as a signal to tune hyperparameters or curate data while preserving the actor-critic architecture. EVPO departs from both strategies by elevating EV from a passive diagnostic to a architecture selection criterion that determines *whether* the critic is used at all, switching to critic-free estimation when the value network becomes mathematically harmful.

6 Conclusion

We presented EVPO, an adaptive RL method for LLM post-training that uses explained variance to switch between critic-based and batch-mean advantage estimation at each training step. By casting baseline selection as a Kalman filtering problem, we proved that the sign of EV is the exact boundary separating the variance-reducing from the variance-inflating critic regime (Theorem 1), and thus enhance the training process at every step.

Across four diversified tasks, EVPO consistently outperforms both fixed baselines. The gating analysis confirms that EVPO's adaptation tracks critic maturation over training, and the threshold sensitivity study validates that the theoretically derived zero boundary is empirically optimal.

Limitations

Population vs. batch EV. Theorem 1 characterizes the collapse boundary in terms of population EV, whereas EVPO switches on batch EV computed from finite samples. We do not provide

finite-sample concentration bounds for the switching rule, though the threshold sensitivity analysis (Section 4.4) suggests that zero is a robust practical threshold.

Resource overhead. EVPO maintains a critic network throughout training, including iterations where the batch-mean mode is selected, and therefore retains the memory footprint of PPO rather than matching fully critic-free methods.

Task coverage. Our evaluation spans four tasks across three domains. Tasks with dense intermediate rewards, learned reward models, or substantially different rollout lengths may exhibit different EV dynamics.

7 Ethics Statement

EVPO is a methodological contribution: an adaptive policy-optimization algorithm that monitors batch-level explained variance to decide whether the learned critic should serve as the advantage baseline. It does not release any new model, dataset, or capability, and only affects how advantages are estimated from existing rollouts. We will release our training code and configurations, which are assembled from publicly available artifacts; we avoid proprietary resources or materials with unclear usage rights, and will perform a license and attribution review prior to release. EVPO is intended to improve the stability and efficiency of reinforcement learning (RL) post-training rather than to elicit harmful behavior or introduce new dangerous capabilities. We utilize the models and datasets in accordance with their original design intentions and targets. None of the data contains personal information.

All of our experiments are conducted on existing, publicly available task suites spanning classical control, agentic interaction, and mathematical reasoning, and our study does not involve recruiting external human subjects or crowdworkers. All experimental design, implementation, and result validation were conducted by the research team as part of internal quality assurance, so the work does not constitute human-subjects experimentation and does not require IRB approval. Models trained with EVPO may still inherit the general misuse risks of large language models, but these risks stem from the underlying base model and reward design rather than from our optimizer.

All reported numbers are produced by our code and verified by the authors. To support reproducibility, we will release our implementation, training configurations, and evaluation scripts.

During the course of this study, we also used generative AI for language polishing, and we carefully reviewed and verified all AI-generated content.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *Preprint*, arXiv:2402.14740.

Oron Anschel, Nir Baram, and Nahum Shimkin. 2017. Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, pages 176–185. PMLR.

Cong Chen, Kaixiang Ji, Hao Zhong, Muzhi Zhu, Anzhou Li, Guo Gan, Ziyuan Huang, Cheng Zou, Jiajia Liu, Jingdong Chen, Hao Chen, and Chunhua Shen. 2025. Gui-shepherd: Reliable process reward and verification for long-sequence gui tasks. *Preprint*, arXiv:2509.23738.

Bryan L. M. de Oliveira, Felipe V. Frujeri, Marcos P. C. M. Queiroz, Luana G. B. Martins, Telma W. de L. Soares, and Luckeciano C. Melo. 2025. Learning without critics? revisiting grpo in classical reinforcement learning environments. *Preprint*, arXiv:2511.03527.

Paolo Dell'Aversana. 2021. The frozen lake problem. an example of optimization policy.

Deming Ding, Shichun Liu, Enhui Yang, Jiahang Lin, Ziying Chen, Shihan Dou, Honglin Guo, Weiyu Cheng, Pengyu Zhao, Chengjun Xiao, Qunhong Zeng, Qi Zhang, Xuanjing Huang, Qidi Xu, and Tao Gui. 2026. Octobench: Benchmarking scaffold-aware instruction following in repository-grounded agentic coding. *Preprint*, arXiv:2601.10343.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 2025. Agentic reinforced policy optimization. In *The Fourteenth International Conference on Learning Representations*.

Shihan Dou, Ming Zhang, Zhangyue Yin, Chenhao Huang, Yujiong Shen, Junzhe Wang, Jiayi Chen, Yuchen Ni, Junjie Ye, Cheng Zhang, Huaibing Xie, Jianglu Hu, Shaolei Wang, Weichao Wang, Yanling Xiao, Yiting Liu, Zenan Xu, Zhen Guo, Pluto Zhou, and 8 others. 2026. Cl-bench: A benchmark for context learning. *Preprint*, arXiv:2602.03587.

Yannis Flet-Berliac and Philippe Preux. 2019. Only relevant information matters: Filtering out noisy samples to boost rl.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. 2025. Reinforce++: Stabilizing critic-free policy optimization with global advantage normalization. *Preprint*, arXiv:2501.03262.

Samyak Jain, Ayush Agrawal, and Navin Goyal. 2025. Towards understanding the optimization landscape of grpo and its variants. In *First Workshop on Foundations of Reasoning in Language Models*.

Andreas Junghanns and Jonathan Schaeffer. 2001. Sokoban: Enhancing general single-agent search methods using domain knowledge. *Artificial Intelligence*, 129(1):219–251.

R. E. Kalman. 1960. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.

Lijun Li, Xiaoyu Wang, Jianan Liu, Haoran Chen, Minghao Zhang, Zixuan Yang, and Haoran Zhang. 2025. Unifying ppo, dpo, and grpo: A theoretical and empirical study on llm post-training.

Jiahang Lin, Kai Hu, Binghai Wang, Yuhao Zhou, Zhiheng Xi, Honglin Guo, Shichun Liu, Junzhe Wang, Shihan Dou, Enyu Zhou, Hang Yan, Zhenhua Han, Tao Gui, Qi Zhang, and Xuanjing Huang. 2026. Mmdoc-r1: Training agents for long document visual question answering through multi-turn reinforcement learning. *Preprint*, arXiv:2604.13579.

Naemeh Mohammadpour, Meysam Fozi, Mohammad Mehdi Ebadzadeh, Ali Azimi, and Ali Kamali. 2024. Proximal policy optimization with adaptive generalized advantage estimate: Critic-aware refinements.

OpenAI, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, and 243 others. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Yan Pei, Swarnendu Biswas, Donald S. Fussell, and Keshav Pingali. 2019. An elementary introduction to kalman filtering. *Preprint*, arXiv:1710.04055.

Qwen, 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, and 24 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents.

John Schulman. 2016. The nuts and bolts of deep rl research.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2018. High-dimensional continuous control using generalized advantage estimation. *Preprint*, arXiv:1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv: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. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. Alfworld: Aligning text and embodied environments for interactive learning. *Preprint*, arXiv:2010.03768.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, A. J. Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, and 465 others. 2025. Openai gpt-5 system card. *Preprint*, arXiv:2601.03267.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press.

Luong Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. Reft: Reasoning with reinforced fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7601–7614, Bangkok, Thailand. Association for Computational Linguistics.

Junzhe Wang, Zhiheng Xi, yajie yang, Hao Luo, Shihan Dou, Tao Gui, and Qi Zhang. 2026. Enhancing llm-based search agents via contribution weighted group relative policy optimization. *Preprint*, arXiv:2604.14267.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. 2025.

Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *Preprint*, arXiv:2504.20073.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Shunyu Yao, Howard Chen, John Yang, and Karthik R. Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu, Cheng Chang, Zhangyue Yin, Rongxiang Weng, and 8 others. 2023. Secrets of rlhf in large language models part i: Ppo. *Preprint*, arXiv:2307.04964.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

A Extended Proofs and Derivations

Detailed proofs and derivations are provided in this appendix.

A.1 Detailed Kalman Filtering Analogy

We provide here the full state estimation analogy and derivation that motivates the EV-based switching criterion introduced in Section 2.3.

State estimation analogy. We cast baseline construction as a scalar state estimation problem. The hidden state to be estimated is the true value function $x = V^\pi(s)$. For each sample in a training batch, the observed return decomposes as

$$G = V^\pi(s) + \epsilon, \quad \epsilon \sim (0, R), \quad R \triangleq \mathbb{E}[\mathrm{Var}(G \mid s)], \tag{10}$$

where R is the irreducible *sampling noise* arising from the stochasticity of future actions under π. Two competing estimators of $V^\pi(s)$ are available at each training step:

- **Model-driven prediction (Critic).** $\hat{x}_A = \hat{V}_\phi(s) = V^\pi(s) + \delta(s)$, where $\delta(s)$ denotes the critic's estimation error with variance $P_A \triangleq \mathrm{Var}(\delta)$. This parallels the *state prediction* in a Kalman filter: it exploits learned state features but carries modelling noise whose magnitude depends on the critic's current training quality.

- **Evidence-driven observation (Batch mean).** $\hat{x}_B = \bar{G}_\mathcal{B}$, the empirical mean return over the batch, which approximates $\mathbb{E}[V^\pi(s)]$. Its error relative to any specific state s has variance $P_B \triangleq \mathrm{Var}_s(V^\pi(s))$, paralleling the *observation noise* in a Kalman filter: it reflects how much true state values vary within the batch, and is large when the batch covers heterogeneous states.

The sampling noise R affects both estimators symmetrically and controls the *sensitivity* of the EV diagnostic without altering its *sign*.

Optimal fusion and the Kalman gain. The Kalman filter combines the two estimators via a linear combination $b(s) = (1 - K)\,\hat{x}_A + K\,\hat{x}_B$ that minimizes the mean squared error $\mathbb{E}[(b(s) - V^\pi(s))^2]$. The optimal *Kalman gain* is:

$$K^* = \frac{P_A}{P_A + P_B}. \tag{11}$$

When $K^* < \frac{1}{2}$ (i.e., $P_A < P_B$), the critic is more reliable than the batch mean and should receive higher weight; when $K^* \geq \frac{1}{2}$ (i.e., $P_A \geq P_B$), the critic's estimation noise dominates the state signal it provides, and the baseline should retreat toward the batch mean.

The Cont. K method. Since both P_A and P_B in the K^* formula depend on the unknown $V^\pi(s)$ and are therefore unobservable, we replace them with samplable surrogates $\tilde{P}_A = \mathrm{Var}(\hat{V}_\phi(s) - G)$

and $\tilde{P}_B = \mathrm{Var}(G)$, yielding the estimate:

$$\begin{aligned}\tilde{K} &= \frac{\tilde{P}_A}{\tilde{P}_A + \tilde{P}_B} \\ &= \frac{\mathrm{Var}(\hat{V}_\phi(s) - V^\pi(s) - \epsilon)}{\mathrm{Var}(V^\pi(s) + \epsilon) + \mathrm{Var}(\hat{V}_\phi(s) - V^\pi(s) - \epsilon)} \\ &= \frac{P_A + R}{P_A + P_B + 2R}.\end{aligned} \tag{12}$$

Using $\tilde{K}$ as the linear combination coefficient at each step, i.e., $b(s) = (1 - \tilde{K})\,\hat{V}_\phi(s) + \tilde{K}\,\bar{G}_{\mathcal{B}}$, constitutes the Cont. K method. Because the return variance R enters both the numerator and denominator of $\tilde{K}$, the estimation target shifts from $\frac{P_A}{P_A+P_B}$ to $\frac{P_A+R}{P_A+P_B+2R}$. While $\tilde{K}$ still lies in $[0, 1]$, it is systematically pulled toward $\frac{1}{2}$.

Connection to explained variance. Substituting the variance decompositions from Eq. (10) into the definition of EV yields:

$$\mathrm{ev} = \frac{P_B - P_A}{P_B + R}. \tag{13}$$

Since the denominator is always positive, the sign of EV is determined solely by $P_B - P_A$:

- $\mathrm{ev} > 0 \iff P_A < P_B \iff K^* < \frac{1}{2}$: the critic outperforms the batch mean, corresponding to the regime where PPO's state-dependent baseline is advantageous;
- $\mathrm{ev} \leq 0 \iff P_A \geq P_B \iff K^* \geq \frac{1}{2}$: the critic's noise exceeds its signal, corresponding to the regime where GRPO's constant baseline is preferable.

A.2 GAE Derivation under RLVR Conditions

Proximal Policy Optimization (Schulman et al., 2017) estimates advantages via Generalized Advantage Estimation (GAE) (Schulman et al., 2018):

$$\begin{aligned}\hat{A}_t^{\mathrm{GAE}} &= \sum_{l=0}^{T-1-t} (\gamma\lambda)^l \delta_{t+l}, \\ \delta_t &= r_t + \gamma \hat{V}_\phi(s_{t+1}) - \hat{V}_\phi(s_t),\end{aligned} \tag{14}$$

where $\hat{V}_\phi$ is a learned critic and $\lambda \in [0, 1]$ controls the bias–variance trade-off. The policy is updated by maximizing a clipped surrogate objective that constrains the ratio $\pi_\theta(a_t|s_t)/\pi_{\theta_{\mathrm{old}}}(a_t|s_t)$.

Under RLVR conditions ($\gamma = \lambda = 1$, terminal-only reward $r_T = \mathcal{R}(x, y)$, $r_t = 0$ for $t < T$), GAE reduces to:

$$\hat{A}_t^{\mathrm{PPO}} = \sum_{l=t}^{T-1} \big(r_l + \hat{V}_\phi(s_{l+1}) - \hat{V}_\phi(s_l)\big) = G - \hat{V}_\phi(s_t), \tag{15}$$

where the telescoping sum cancels all intermediate critic terms (with the terminal state value defined as 0), leaving the advantage as the deviation of the observed return from the critic's state-dependent prediction.

B Tasks

We evaluate on four tasks spanning multi-turn grid-world control, interactive web agents, and single-turn mathematical reasoning. All training code is based on the `RAGEN2` framework (Wang et al., 2025), which wraps `verl` for PPO-style LLM post-training and provides multi-turn agent environments.

Sokoban. We use the `CoordSokoban` variant of the RAGEN2 Sokoban environment (Junghanns and Schaeffer, 2001). Each episode instantiates a 6×6 grid with one box and one target, procedurally generated from an environment seed. The observation at each step is a symbol grid together with zero-indexed coordinates of the player, the box, and the target. The action space is {UP, DOWN, LEFT, RIGHT}; the policy is prompted to emit an action sequence of length up to two per turn, separated by "`||`" and enclosed in `<answer>. . . </answer>` tags. A trajectory consists of up to 10 executed actions across up to 5 turns. The reward is terminal and binary: $+1$ if all boxes end on targets, 0 otherwise, with an additional format penalty of -0.1 for malformed outputs.

FrozenLake. We use the `CoordFrozenLake` variant wrapping the Gymnasium `FrozenLake-v1` environment (Dell'Aversana, 2021). The map is a 4×4 grid with randomly placed holes, the start tile, and the goal tile; the map layout is resampled per environment seed. Transitions are stochastic (`is_slippery=True`): the commanded action succeeds with probability $1/3$, and with probability $2/3$ the agent slips perpendicular to the intended direction. The observation is again a symbol grid plus coordinates of start, goal, player, and all holes. The action space is {LEFT, DOWN, RIGHT, UP}, with up to 10 actions per trajectory across up to 5 turns. The reward is terminal and binary: $+1$ if the

player reaches the goal, 0 if the player falls into a hole or exhausts the action budget; a format penalty of -0.1 is applied for malformed outputs.

WebShop. WebShop (Yao et al., 2022) is an interactive web-navigation task in which the agent must locate and purchase a product matching a natural-language instruction. We use the `webshop-minimal` fork with the *small* product catalogue shipped with RAGEN2. At each turn the agent observes the current page (search box, product list, or product detail) together with the list of clickable elements, and produces exactly one action drawn from `search[keywords]` or `click[element]`, again wrapped in `<answer>. . . </answer>`. A trajectory consists of up to 9 actions across up to 9 turns. The reward is the built-in WebShop matching score in $[0, 1]$, delivered only at the terminal `click[buy now]`.

MATH. We use DAPO-Math-17k (Yu et al., 2025) (17,267 competition-level problems) for *both* training and validation: the full training set is used for RL updates, and a *fixed subset of 32 problems*, held out from training, is used for evaluation. Each problem is a single-turn reasoning task (`max_turn=1`): the policy is prompted with the problem statement and must emit a `<think>. . . </think>` block followed by the final numerical answer inside `<answer>. . . </answer>`. The reward is terminal and binary: $+1$ if the final answer matches the ground truth (after normalization), 0 otherwise.

C Baselines and Hyperparameters

We compare EVPO against a Base-LLM reference and four RL baselines that span both critic-based and critic-free families.

- **Base LLM.** Instruction-tuned Qwen2.5 without any RL post-training, prompted zero-shot with the same task template as the RL runs.
- **PPO** (Schulman et al., 2017). Standard clipped-surrogate PPO with GAE advantages. Under our RLVR setting ($\gamma{=}\lambda{=}1$, terminal-only reward) this reduces to the PPO advantage of Eq. (1) with a learned, state-dependent critic.
- **StarPO-S** (Wang et al., 2025). A multi-turn PPO variant introduced by RAGEN that filters the lowest-variance rollouts before each update (`rollout_filter_ratio`$=0.25$, `rollout_filter_metric`=reward_variance) to stabilize long-horizon agentic training.
- **GRPO** (Shao et al., 2024). Critic-free; the baseline is the response-level batch mean, normalized by the batch standard deviation.
- **DAPO** (Yu et al., 2025). A modern critic-free variant built on GRPO with (i) dynamic sampling that discards groups whose rewards are all identical and resamples up to `max_num_gen_batches`$=5$ times until enough informative groups are collected; (ii) *clip-higher* asymmetry $(\varepsilon_{\text{low}}, \varepsilon_{\text{high}}) = (0.2, 0.28)$; (iii) token-level loss aggregation (`loss_agg_mode`=token-mean); and (iv) no std normalization of advantages and no in-reward KL term.

All five methods (PPO, StarPO-S, GRPO, DAPO, EVPO) share the same rollout stack, tokenizer, environment code and 8×H100 GPU server, and differ only in the advantage estimator and the listed hyperparameters. EVPO itself uses PPO-style training (critic and actor with clipped surrogate) and only changes the baseline subtracted from the return, as prescribed by the EV-gating rule (Eq. (8)).

Table 2 lists the hyperparameters that differ across methods or tasks. Learning rates are tuned per task on the Base-LLM vs. PPO gap on a single seed and then *kept identical across methods on the same task* for fair comparison, except where a differing value is itself a defining feature of the algorithm (see the caption of Table 2).

D Details of the Preliminary Experiments

This section gives the exact protocols for the two preliminary studies that motivate EVPO in Section 2.

D.1 PPO on Sokoban: EV, Gradient Norm, Success Rate (Figure 2)

We train a PPO agent on Sokoban using the same setting as the Sokoban PPO main experiment (Sokoban PPO row of Table 2). At every step we record, over the same batch of $M = 16 \times 8 = 128$ rollouts used for the update: (i) the batch-level EV $\widehat{\mathrm{ev}}_{\mathcal{B}}$ defined in Eq. (6); (ii) the pre-clipping gradient norm of the actor; (iii) the training success rate (fraction of rollouts with terminal reward $= 1$).

Table 2: Per-task / per-method hyperparameter overrides. $\varepsilon_{\text{lo}}/\varepsilon_{\text{hi}}$ are the PPO-style clip thresholds. Where DAPO and StarPO-S show different values from the other methods on the same task (e.g., DAPO's clip-higher asymmetry, or the StarPO-S Sokoban configuration that preserves the original RAGEN recipe), those settings are defining features of the respective algorithms rather than tuning choices, and we therefore do not force them to match the other methods.

Task	Method	lr_{actor}	$\text{lr}_{\text{critic}}$	$\varepsilon_{\text{lo}}/\varepsilon_{\text{hi}}$	KL coef.	Steps
Sokoban (6×6)	PPO	4e−6	1e−5	0.20/0.28	0	500
	StarPO-S	1e−6	1e−5	0.20/0.28	0	400
	GRPO	4e−6	–	0.20/0.28	0	400
	DAPO	4e−6	–	0.20/0.28	0	400
	EVPO	4e−6	1e−5	0.20/0.28	0	500
FrozenLake (4×4)	PPO	1e−6	1e−5	0.20/0.28	0	200
	StarPO-S	1e−6	1e−5	0.20/0.28	0	200
	GRPO	1e−6	–	0.20/0.28	0	200
	DAPO	1e−6	–	0.20/0.28	0	200
	EVPO	1e−6	1e−5	0.20/0.28	0	200
WebShop	PPO	5e−7	5e−6	0.20/0.20	10^{-2}	100
	StarPO-S	1e−6	1e−5	0.20/0.28	0	100
	GRPO	5e−7	–	0.20/0.20	10^{-2}	100
	DAPO	5e−7	–	0.20/0.28	10^{-2}	100
	EVPO	5e−7	5e−6	0.20/0.20	10^{-2}	100
MATH	PPO	2e−6	1e−5	0.20/0.20	0	200
	StarPO-S	2e−6	1e−5	0.20/0.28	0	200
	GRPO	2e−6	–	0.20/0.20	0	200
	DAPO	2e−6	–	0.20/0.28	0	200
	EVPO	2e−6	1e−5	0.20/0.20	0	200

Shaded regions in Figure 2(a–c) highlight the first ≈ 150 steps where $\widehat{\text{ev}}_{\mathcal{B}}$ remains near or below 0.

Noise-injection experiment (Figure 2d). We first train a PPO agent on FrozenLake to convergence under the same hyperparameters as the FrozenLake row of Table 2 for 200 steps, then continue training with i.i.d. Gaussian noise added to every critic output: $\hat{V}_\phi(s) \mapsto \hat{V}_\phi(s) + \xi$ with $\xi \sim \mathcal{N}(0, \sigma_\delta^2)$. We sweep $\sigma_\delta \in \{0.3,\ 1,\ 3,\ 10\}$, corresponding to noise magnitudes from well below to well above the mean absolute state value (≈ 0.5). Each noise level is run for an additional 150 training steps, starting from the same converged checkpoint. We report the smoothed validation success rate; Fig. 2d shows monotone degradation with complete collapse at σ_δ=10, precisely the regime where $\widehat{\text{ev}}_{\mathcal{B}}$ is driven far below zero.

D.2 Cold-start and Critic Warmup (Figure 6)

We run two interventions on top of the Sokoban and FrozenLake PPO setups.

Cold-start. During the first K_{cold} training steps we replace every critic output in the advantage computation by the batch mean, $\hat{A}_t^{\text{cold-start}} = G - \bar{G}_{\mathcal{B}}$; after step K_{cold} we resume standard PPO with the original critic. We set $K_{\text{cold}} = 50$ on Sokoban and $K_{\text{cold}} = 25$ on FrozenLake, chosen to cover the EV-negative phase of Fig. 2a. The critic itself is always trained with its usual MSE loss so that by step K_{cold} it has had time to warm up.

Critic warmup. Before the actor receives any gradient, we train the critic alone for $K_{\text{warm}} = 200$ steps using the same rollouts as full PPO but with $\nabla_\theta \mathcal{L}_{\text{actor}}$ set to zero. After step K_{warm} we switch on actor updates and continue with standard PPO.

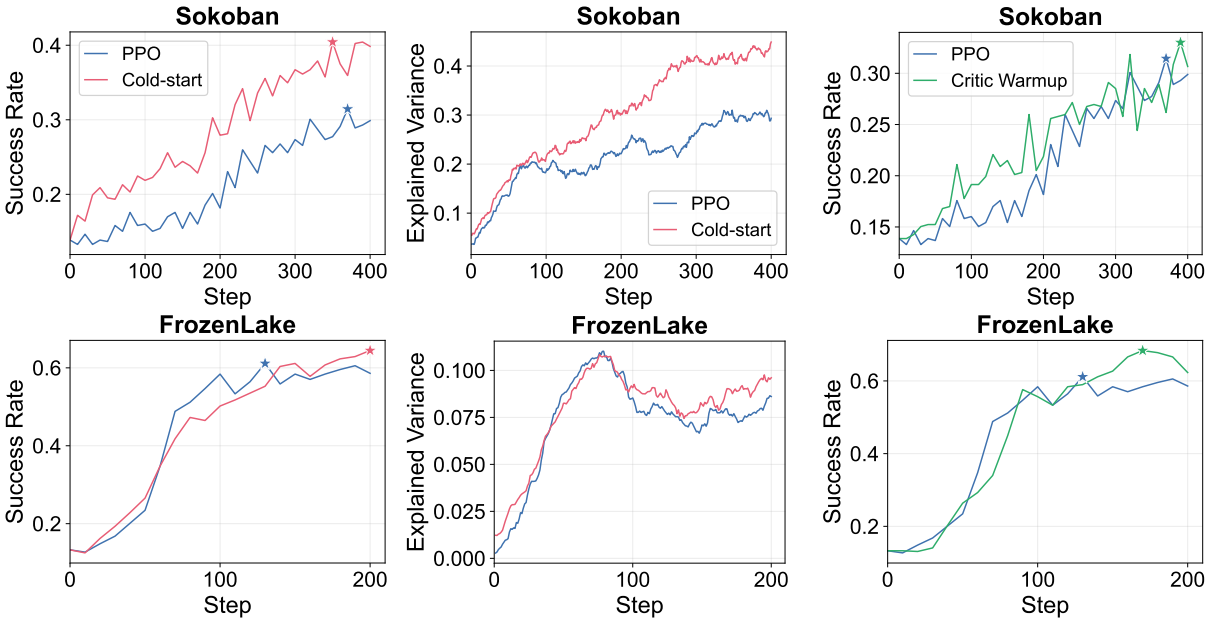
In both interventions all other hyperparameters match the PPO rows of Table 2. Figure 6 compares each intervention against the unmodified PPO baseline under the same seed and hardware; the consistent improvements confirm that it is specifically the suppression of critic-driven variance (rather than any secondary effect of the interventions) that benefits the policy.

E Additional Experiments

To further validate the effectiveness of EVPO, we present additional ablation and robustness analyses.

E.1 Gating Mechanism Ablation

Although EVPO is a simple algorithm, the batch-mean mode it gates into differs subtly from the GRPO baseline: it uses the unnormalized batch



(a) Cold-start (b) Critic Warmup

Figure 6: Reducing critic variance helps the actor. **(a)** Cold-start on Sokoban (top row) and FrozenLake (bottom row): replacing the critic output with $\bar{G}_{\mathcal{B}}$ during early training improves both the success rate (left column) and the explained variance of the critic (right column), confirming that the intervention raises critic quality precisely in the regime where it was initially harmful. **(b)** Critic Warmup: pre-training the critic before actor training begins, on Sokoban (top) and FrozenLake (bottom).

mean $\bar{G}_{\mathcal{B}}$ rather than GRPO's std-normalized advantages, and the critic continues to be trained even when its output is not used for the actor. To isolate the contribution of the gating mechanism itself, we compare EVPO against its two constituent modes run in isolation across all four tasks:

- **Critic Mode**: the actor always uses the critic baseline $b(s_t) = \hat{V}_\phi(s_t)$, equivalent to standard PPO under our RLVR setting.
- **Batch-Mean Mode**: the actor always uses $b(s_t) = \bar{G}_{\mathcal{B}}$.

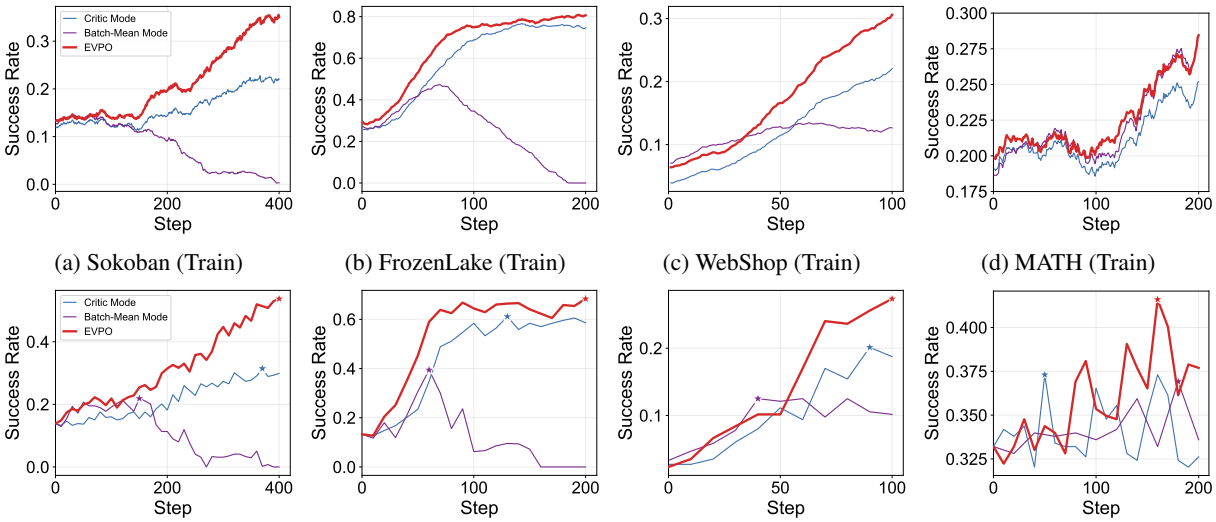
As shown in Figure 7, neither fixed mode dominates throughout training. Batch-Mean Mode provides faster initial progress (when the critic is still unreliable), while Critic Mode eventually surpasses it once the critic matures. EVPO inherits the best of both: it matches Batch-Mean Mode's early acceleration and transitions to Critic Mode's sustained improvement, confirming that the EV-gating mechanism successfully identifies and exploits the crossover point between the two regimes.

E.2 Robustness to Group Size

To test whether EVPO remains effective under different batch configurations, we reduce the training group size from $M{=}16$ to $M{=}8$ (i.e., each prompt is sampled 8 times per step instead of 16) and re-run PPO, GRPO, and EVPO on Sokoban and FrozenLake. All other hyperparameters are kept identical to the main experiments (Table 2).

As shown in Figure 8, EVPO retains its advantage over PPO and GRPO across all four tasks even with halved group size, confirming that the binary EV diagnostic remains a reliable switching signal under noisier variance estimates.

Furthermore, we compare the gating frequency (proportion of steps using the critic) between $M{=}8$ and $M{=}16$ in Figure 9. Under the smaller group size, the critic-usage rate decreases consistently across tasks. This is expected from our theoretical framework: with fewer samples per group, the batch-level EV estimate $\widehat{\mathrm{ev}}_{\mathcal{B}}$ has higher variance and is more likely to fall below zero, causing the gate to select the batch-mean mode more often. Notably, the anomalous gating-frequency pattern previously observed on FrozenLake under $M{=}16$ also disappears at $M{=}8$, suggesting that it was an artifact of the larger batch rather than a failure of the gating mechanism. The fact that EVPO adapts its gating frequency in a theoretically predictable manner—rather than breaking down—provides additional evidence that the EV-based switching cri-



(a) Sokoban (Train) (b) FrozenLake (Train) (c) WebShop (Train) (d) MATH (Train)

(e) Sokoban (Val) (f) FrozenLake (Val) (g) WebShop (Val) (h) MATH (Val)

Figure 7: Gating mechanism ablation. Top row: training success rate; bottom row: validation success rate. EVPO inherits the fast early growth of Batch-Mean Mode and the sustained later improvement of Critic Mode.

terion is well-calibrated.

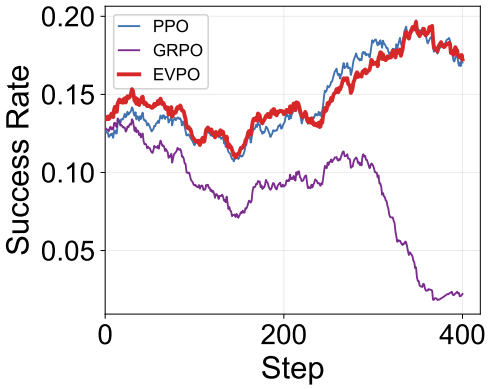
F Ethics Statement

EVPO is a methodological contribution: an adaptive policy-optimization algorithm that monitors batch-level explained variance to decide whether the learned critic should serve as the advantage baseline. It does not release any new model, dataset, or capability, and only affects how advantages are estimated from existing rollouts. We will release our training code and configurations, which are assembled from publicly available artifacts; we avoid proprietary resources or materials with unclear usage rights, and will perform a license and attribution review prior to release. EVPO is intended to improve the stability and efficiency of reinforcement learning (RL) post-training rather than to elicit harmful behavior or introduce new dangerous capabilities. We utilize the models and datasets in accordance with their original design intentions and targets. None of the data contains personal information.

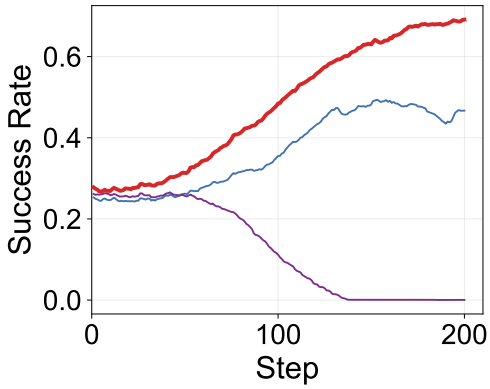
All of our experiments are conducted on existing, publicly available task suites spanning classical control, agentic interaction, and mathematical reasoning, and our study does not involve recruiting external human subjects or crowdworkers. All experimental design, implementation, and result validation were conducted by the research team as part of internal quality assurance, so the work does not constitute human-subjects experimentation and does not require IRB approval. Models trained with EVPO may still inherit the general misuse risks of large language models, but these risks stem from the underlying base model and reward design rather than from our optimizer.

All reported numbers are produced by our code and verified by the authors. To support reproducibility, we will release our implementation, training configurations, and evaluation scripts.

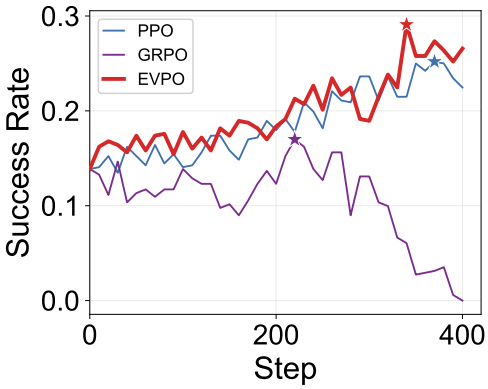
During the course of this study, we also used generative AI for language polishing, and we carefully reviewed and verified all AI-generated content.



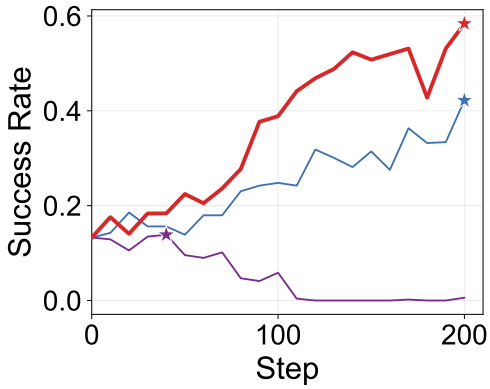
(a) Sokoban (Train)



(b) FrozenLake (Train)

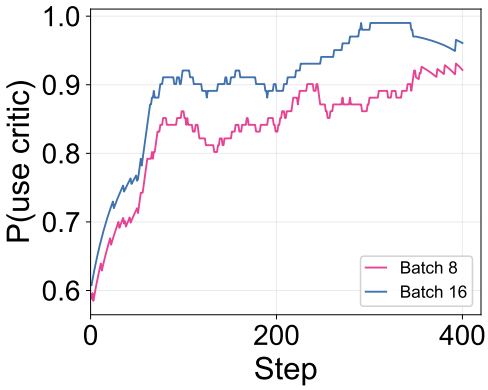


(c) Sokoban (Val)

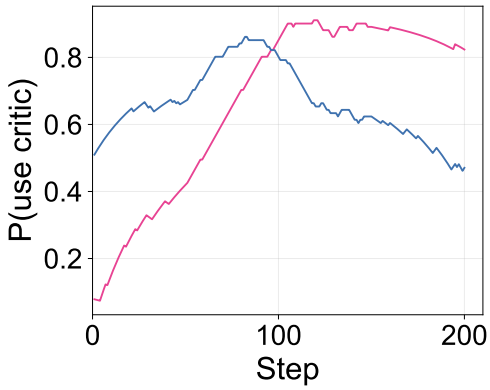


(d) FrozenLake (Val)

Figure 8: Robustness to group size (M=8). Top row: training success rate; bottom row: validation success rate. EVPO consistently outperforms both PPO and GRPO under the reduced group size, demonstrating that the EV-gating mechanism does not rely on large-batch variance estimates to function effectively.



(a) Sokoban



(b) FrozenLake

Figure 9: Gating frequency comparison between group sizes. Batch 8 vs. Batch 16. Under the smaller group size, EVPO gates into critic mode less frequently, consistent with the higher estimation noise in $\widehat{\mathrm{ev}}_{\mathcal{B}}$ when fewer samples are available.