LEARNING WHAT THE GROUP CANNOT TELL: DECOMPOSED CRITICS FOR LLM REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) has become an important approach for post-training large language models (LLMs). In many LLM RL tasks, feedback is sparse or available only at the end of a trajectory, making credit assignment across intermediate states important for effective policy optimization. Proximal Policy Optimization (PPO) addresses this problem with a learned value function, which estimates the expected return of intermediate states and provides state-dependent advantages for policy updates. However, modern LLM-based RL algorithms typically collect multiple rollouts for each prompt, thereby exposing group-level information that conventional critics do not explicitly exploit. In this work, we revisit critic learning in such grouped settings. Instead of learning the full state value directly from individual terminal outcomes, we decompose value estimation into two complementary components: a prompt-level success prior estimated from grouped rollouts and history-dependent evidence learned by the critic. Based on this decomposition, we introduce **G**roup-**A**nchored **V**alue **E**stimation (GAVE), which trains the critic with a group-conditional objective and reconstructs standard state values by combining the learned correction with an empirical prompt prior. We further prove that estimation errors in the empirical prior are not amplified in expectation during reconstruction, yielding a finite-sample error bound. Experiments on LLM agent tasks show that GAVE consistently improves both overall and within-prompt value estimation compared with conventional critics and information-matched baselines, while also leading to improved policy optimization.

1 INTRODUCTION

Reinforcement learning (RL) has become a key paradigm for post-training large language models (LLMs) on complex reasoning and interactive tasks, including mathematical problem solving, code generation, web navigation, and tool use (OpenAI et al., 2024; DeepSeek-AI, 2025; Yao et al., 2022; Jin et al., 2025).In these settings, models often make a sequence of intermediate decisions before receiving feedback on the final outcome. However, these decisions can differ substantially in their contribution on success: some are critical, while others may be irrelevant or even detrimental. Effective policy optimization therefore requires assigning credit to individual decisions, which becomes difficult when feedback is available only after a long delay, often in the form of a sparse terminal reward.

Proximal Policy Optimization (PPO) addresses this credit-assignment problem with a learned value function that estimates the expected return of intermediate states, which GAE uses to construct step-level advantage estimates (Schulman et al., 2017; 2016). When supervision is available only at the end of a trajectory, however, the critic must learn these values from terminal outcomes alone, making accurate value estimation difficult. VinePPO (Kazemnejad et al., 2025) addresses this issue by estimating state values from multiple continuations branched from intermediate states, thereby providing more direct supervision for value estimation. This yields more informative value targets, but requires additional rollouts specifically for estimating them.

An alternative line of work dispenses with a learned value function and instead estimates advantages directly from groups of responses to the same prompt. RLOO (Ahmadian et al., 2024) and GRPO (Shao et al., 2024), for example, construct advantage estimates from within-group reward statistics rather than a learned value baseline. Grouped rollouts are also increasingly common in

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.

PPO-style LLM training. However, the critic typically does not make use of this group structure. Instead, the rollouts are treated as independent training examples, with each state supervised only by the return of its own trajectory. This raises a natural question: *can the information available within a rollout group be used to improve how the critic is learned?*

Following this observation, we reconsider the learning target of the critic. For binary terminal rewards, the state value admits a Bayesian decomposition into a prompt-level success prior and history-dependent evidence. We therefore use grouped outcomes to estimate the former and train the critic only on the latter. This leads to **Group-Anchored Value Estimation (GAVE)**, a group-aware framework for critic learning. GAVE learns the history-dependent evidence through a group-conditional objective and combines it with a leave-one-out estimate of the prompt prior to recover the standard on-policy value.

We evaluate GAVE on multi-turn agent tasks, including ALFWorld, WebShop, and Search QA, and on a suite of mathematical reasoning benchmarks. Our experiments show that GAVE consistently provides more accurate value estimates than conventional critics and information-matched baselines, while also improving downstream PPO performance. Further analyses show stronger within-prompt value estimation, indicating that the gains extend beyond overall prediction accuracy. These results show that grouped rollout information can be effectively incorporated into critic learning to improve both value estimation and policy optimization.

2 PRELIMINARIES

2.1 LLM REINFORCEMENT LEARNING AS AN MDP

We formulate reinforcement learning for language models as a finite-horizon Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $\mathcal{P}$ the transition dynamics, r the reward function, and γ the discount factor. Given an input $x \sim \mathcal{D}$, a policy π_θ induces a trajectory $\tau = (s_1, a_1, \ldots, s_T, a_T)$, where s_t denotes the information available before decision t and $a_t \sim \pi_\theta(\cdot \mid s_t)$ the corresponding action. The next state follows $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$, with reward $r_t = r(s_t, a_t)$.

Following common practice in LLM reinforcement learning, we set $\gamma = 1$ and omit it from subsequent notation unless stated otherwise. The value function under policy π is defined as $V^\pi(s_t) = \mathbb{E}_\pi \left[\sum_{t'=t}^{T} r_{t'} \,\middle|\, s_t \right]$, representing the expected return from state s_t. The RL objective is to maximize the expected cumulative reward

$$J(\theta) = \mathbb{E}_{x \sim \mathcal{D},\, \tau \sim \pi_\theta(\cdot|x)} \left[\sum_{t=1}^{T} r_t \right]. \tag{1}$$

In this work, we focus on settings where supervision is provided through a terminal outcome $R(x, \tau)$, i.e., $r_t = 0$ for $t < T$ and $r_T = R(x, \tau)$.

2.2 PROXIMAL POLICY OPTIMIZATION

Proximal Policy Optimization (PPO) (Schulman et al., 2017) updates the policy using a clipped surrogate objective. Given trajectories sampled from the behavior policy π_{old}, PPO maximizes

$$\mathcal{J}_{\text{PPO}}(\theta) = \mathbb{E}\Big[\min\Big(\rho_t(\theta) \widehat{A}_t, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\, \widehat{A}_t \Big) \Big], \tag{2}$$

where $\rho_t(\theta) \coloneqq \pi_\theta(a_t \mid s_t) / \pi_{\text{old}}(a_t \mid s_t)$ is the importance ratio, $\widehat{A}_t$ is the advantage estimate, and ϵ is the clipping parameter.

PPO commonly computes $\widehat{A}_t$ using Generalized Advantage Estimation (GAE) (Schulman et al., 2016). Given a value function V, the temporal-difference residual and the resulting advantage estimate are

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t), \qquad \widehat{A}_t = \sum_{\ell=0}^{T-t} (\gamma\lambda)^\ell \delta_{t+\ell}. \tag{3}$$

The value function is typically approximated by a learned critic V_ϕ, which is trained to minimize the discrepancy between its predictions and target returns, commonly using a Mean Squared Error (MSE) loss:

$$\mathcal{L}_{\text{critic}}(\phi) = \mathbb{E}_t \left[\left(V_\phi(s_t) - R_t \right)^2 \right], \tag{4}$$

where $V_\phi(s_t)$ denotes the predicted value of state s_t parameterized by ϕ, and R_t denotes the target return.

3 METHOD

We introduce GAVE, which estimates prompt priors from grouped outcomes and learns history-dependent evidence with a neural critic. A Bayesian decomposition relates these two components to the state value (Section 3.1). We use this decomposition to derive a group-conditional critic objective and reconstruct state values for PPO (Section 3.2). We then bound the reconstruction error due to prior estimation and evidence approximation (Section 3.3).

3.1 PROMPT PRIORS AND HISTORY EVIDENCE

We begin by revisiting the value function under binary terminal rewards. In particular, we consider how the success probability at an intermediate state relates to the success probability of the prompt before any interaction history is observed. This relation reveals a prompt-level prior and a history-dependent term that together determine the state value.

For a prompt x, its success probability under policy π is $p^\pi(x) := P^\pi(R = 1 \mid x)$. At state $s_t = (x, y_{<t})$, the value $V^\pi(s_t) = P^\pi(R = 1 \mid x, y_{<t})$ is the posterior success probability after observing the history $y_{<t}$. By Bayes' rule,

$$V^\pi(s_t) = \frac{P^\pi(y_{<t} \mid x, R = 1)\, p^\pi(x)}{P^\pi(y_{<t} \mid x)}, \qquad 1 - V^\pi(s_t) = \frac{P^\pi(y_{<t} \mid x, R = 0)\, (1 - p^\pi(x))}{P^\pi(y_{<t} \mid x)}. \tag{5}$$

Taking the ratio eliminates the marginal likelihood and gives

$$\frac{V^\pi(s_t)}{1 - V^\pi(s_t)} = \frac{p^\pi(x)}{1 - p^\pi(x)} \frac{P^\pi(y_{<t} \mid x, R = 1)}{P^\pi(y_{<t} \mid x, R = 0)}. \tag{6}$$

Taking logarithms yields the decomposition

$$\operatorname{logit} V^\pi(s_t) = \operatorname{logit} p^\pi(x) + \underbrace{\log \frac{P^\pi(y_{<t} \mid x, R = 1)}{P^\pi(y_{<t} \mid x, R = 0)}}_{\Delta^\pi(s_t)}, \tag{7}$$

where $\operatorname{logit}(v) = \log(v/(1-v))$ and $\Delta^\pi(s_t)$ is the log-likelihood ratio of the observed history under successful and unsuccessful trajectories.

The prior log-odds are shared across rollouts from the same prompt, while $\Delta^\pi(s_t)$ captures the change associated with the observed history. This term is positive when the history is more likely under successful trajectories and negative when it is more likely under unsuccessful ones. Thus, the decomposition separates the initial odds of success from the evidence accumulated along an individual trajectory.

3.2 GROUP-CONDITIONAL EVIDENCE CRITIC

Grouped rollouts allow us to learn the evidence term in Eq. (7) without explicitly estimating the prompt success probability. We obtain the training objective by conditioning on the number of successful rollouts from each prompt.

For each prompt x, we sample G independent rollouts $y^{(1)}, \ldots, y^{(G)} \sim \pi(\cdot \mid x)$. The i-th rollout has terminal outcome $R_i = R(x, y^{(i)})$, and $K = \sum_{j=1}^{G} R_j$ denotes the number of successful rollouts. At step t, we write $s_{i,t} = (x, y^{(i)}_{<t})$. The group-conditioned success probability can be expressed in terms of the same history evidence $\Delta^\pi(s_{i,t})$:

Proposition 1 (Group-conditional evidence). *Suppose $K = k$ for some $k \in \{0, \dots, G\}$. Then, for any rollout i and step t,*

$$P^\pi(R_i = 1 \mid s_{i,t}, K = k) = \sigma\left(\log \frac{k}{G-k} + \Delta^\pi(s_{i,t})\right), \tag{8}$$

with the convention $\log \frac{k}{G-k} = -\infty$ for $k = 0$ and $+\infty$ for $k = G$.

The proof is given in Appendix A.1. Equation (8) does not depend on the prompt success probability $p^\pi(x)$. We therefore define an estimator $f_\phi : \mathcal{S} \to \mathbb{R}$ of Δ^π and minimize

$$\mathcal{L}_{\text{critic}}(\phi) = \mathbb{E}\left[\ell_{\text{BCE}}\left(f_\phi(s_{i,t}) + \log \frac{K}{G-K}, R_i\right)\right], \tag{9}$$

where ℓ_{BCE} denotes binary cross-entropy applied to logits.

For policy optimization, we combine the learned evidence with a leave-one-out estimate of the prompt prior. To keep the empirical prior in $(0, 1)$ when computing its log-odds, we use symmetric pseudocount smoothing,

$$\widetilde{p}_{-i}(x) = \frac{\sum_{j \neq i} R_j + \alpha}{G - 1 + 2\alpha}, \qquad \alpha > 0. \tag{10}$$

Combining this prior with the learned evidence gives the state-value estimator

$$V_\phi(s_{i,t}) = \sigma(\operatorname{logit} \widetilde{p}_{-i}(x) + f_\phi(s_{i,t})). \tag{11}$$

The reconstructed value $V_\phi(s_{i,t})$ serves as the value estimate for policy optimization. Using V_ϕ, we compute the generalized advantage estimates $\widehat{A}_{i,t}$ according to Eq. (3), which are then used in the PPO objective in Eq. (2).

3.3 Stability of Value Reconstruction

We analyze how errors in the prompt prior and history evidence affect the reconstructed value. For a prompt x, write $p = p^\pi(x)$ and define the reconstruction map

$$g(q, \delta) \coloneqq \frac{qe^\delta}{1 - q + qe^\delta}, \qquad q \in [0, 1],\ \delta \in \mathbb{R}. \tag{12}$$

By Eq. (7), the exact value satisfies $V^\pi(s) = g(p, \Delta^\pi(s))$. We next consider the reconstruction obtained from a prior estimate $\widetilde{p}$ and the learned evidence f_ϕ.

Proposition 2 (Reconstruction stability). *Given a prompt-prior estimate $\widetilde{p} \in [0, 1]$ and an evidence estimator $f_\phi : \mathcal{S} \to \mathbb{R}$, the expected reconstruction error satisfies*

$$\mathbb{E}_{s \sim P^\pi(\cdot|x)} \left| g(\widetilde{p}, f_\phi(s)) - V^\pi(s) \right| \leq |\widetilde{p} - p| + \frac{1}{4} \mathbb{E}_{s \sim P^\pi(\cdot|x)} \left| f_\phi(s) - \Delta^\pi(s) \right|. \tag{13}$$

The proof is given in Appendix A.2. In particular, if the history evidence is exact, the expected error of the reconstructed value is no larger than the error in the prompt-prior estimate. More generally, Proposition 2 separates the contributions of prompt-prior estimation and evidence approximation.

To analyze the leave-one-out estimator, consider a group of $m + 1$ independent rollouts from prompt x. We use the outcomes $R_1, \dots, R_m$ of m rollouts to estimate the prior for a state s on the remaining rollout. The smoothed prior and reconstructed value are

$$\widetilde{p}_m = \frac{\sum_{j=1}^m R_j + \alpha}{m + 2\alpha}, \qquad \widehat{V}_m(s) = g(\widetilde{p}_m, f_\phi(s)), \tag{14}$$

where $\alpha > 0$ is the smoothing coefficient from Eq. (10). Writing $\varepsilon_\phi \coloneqq \mathbb{E}_{s \sim P^\pi(\cdot|x)} |f_\phi(s) - \Delta^\pi(s)|$, we obtain the following bound.

Corollary 1 (Finite-sample reconstruction). *The expected value prediction error satisfies*

$$\mathbb{E}\left|\widehat{V}_m(s) - V^\pi(s)\right| \leq \frac{1}{4}\varepsilon_\phi + \frac{\sqrt{mp(1-p) + \alpha^2(1-2p)^2}}{m + 2\alpha}. \tag{15}$$

The expectation is over the sibling rollout outcomes and the state s.

The proof is given in Appendix A.3. The finite-sample term accounts for both sampling variance and the bias introduced by smoothing. For the leave-one-out estimator, $m = G - 1$. With α held constant, this term is $\mathcal{O}(G^{-1/2})$, while the contribution from evidence approximation is bounded by $\varepsilon_\phi/4$.

Table 1: Performance on ALFWorld and WebShop. For ALFWorld, we report success rates (%) for each task type and overall; for WebShop, we report average scores and success rates (%). RL results are reported as means with standard deviations in subscripts. Within each Qwen model group, the highest mean for each metric is in bold and the second-highest is underlined.

Type	Method	ALFWorld							WebShop	
		Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ.
Closed-Source Model										
Prompting	GPT-4o	75.3	60.8	31.2	56.7	21.6	49.8	48.0	31.8	23.7
Prompting	Gemini-2.5-Pro	92.8	63.3	62.1	69.0	26.6	58.7	60.3	42.5	35.9
Qwen3-4B										
RL Training	PPO (MSE)	$70.3_{\pm 2.2}$	$52.0_{\pm 7.3}$	$63.2_{\pm 4.3}$	$73.7_{\pm 2.9}$	$71.1_{\pm 3.9}$	$48.2_{\pm 3.1}$	$63.9_{\pm 1.9}$	$69.1_{\pm 0.7}$	$63.0_{\pm 2.7}$
RL Training	PPO (BCE)	$74.7_{\pm 6.0}$	$39.4_{\pm 4.2}$	$62.0_{\pm 3.5}$	$71.0_{\pm 3.5}$	$73.0_{\pm 3.9}$	$\underline{55.5}_{\pm 5.2}$	$63.6_{\pm 2.6}$	$80.8_{\pm 4.1}$	$69.8_{\pm 1.6}$
RL Training	GRPO	$69.9_{\pm 5.3}$	$56.7_{\pm 6.9}$	$73.6_{\pm 1.3}$	$80.3_{\pm 3.0}$	$\underline{76.8}_{\pm 2.7}$	$11.1_{\pm 4.3}$	$64.1_{\pm 1.0}$	$78.5_{\pm 4.8}$	$69.0_{\pm 3.2}$
RL Training	GiGPO	$\mathbf{82.6_{\pm 4.6}}$	$\underline{80.2}_{\pm 5.0}$	$\underline{82.9}_{\pm 2.0}$	$\underline{84.1}_{\pm 5.3}$	$73.2_{\pm 3.4}$	$36.7_{\pm 2.7}$	$\underline{75.3}_{\pm 1.2}$	$\underline{83.9}_{\pm 3.0}$	$\underline{73.4}_{\pm 4.8}$
RL Training	**GAVE (Ours)**	$\underline{82.1}_{\pm 5.7}$	$\mathbf{97.6_{\pm 3.1}}$	$\mathbf{85.1_{\pm 2.6}}$	$\mathbf{91.6_{\pm 3.7}}$	$\mathbf{86.5_{\pm 6.3}}$	$\mathbf{94.2_{\pm 2.9}}$	$\mathbf{88.6_{\pm 1.4}}$	$\mathbf{89.4_{\pm 1.7}}$	$\mathbf{82.3_{\pm 2.3}}$
Qwen3-8B										
RL Training	PPO (MSE)	$63.9_{\pm 4.8}$	$\underline{81.2}_{\pm 3.3}$	$51.1_{\pm 5.6}$	$66.2_{\pm 3.6}$	$46.1_{\pm 6.2}$	$66.7_{\pm 6.8}$	$60.8_{\pm 2.3}$	$72.6_{\pm 4.4}$	$67.4_{\pm 2.3}$
RL Training	PPO (BCE)	$68.4_{\pm 3.6}$	$46.3_{\pm 14.5}$	$60.0_{\pm 3.0}$	$71.8_{\pm 2.1}$	$52.4_{\pm 6.4}$	$61.9_{\pm 11.5}$	$60.3_{\pm 3.1}$	$83.4_{\pm 3.1}$	$73.7_{\pm 2.4}$
RL Training	GRPO	$74.3_{\pm 3.3}$	$\mathbf{81.5_{\pm 5.5}}$	$78.2_{\pm 3.4}$	$\mathbf{86.6_{\pm 4.4}}$	$58.7_{\pm 6.1}$	$64.6_{\pm 7.4}$	$74.6_{\pm 2.0}$	$81.2_{\pm 4.9}$	$74.2_{\pm 4.3}$
RL Training	GiGPO	$\mathbf{87.8_{\pm 5.3}}$	$74.2_{\pm 9.1}$	$\underline{88.5}_{\pm 2.5}$	$\underline{83.9}_{\pm 3.6}$	$\underline{84.5}_{\pm 2.3}$	$\underline{69.9}_{\pm 5.9}$	$\underline{82.5}_{\pm 1.7}$	$\underline{85.3}_{\pm 3.5}$	$\underline{76.8}_{\pm 2.4}$
RL Training	**GAVE (Ours)**	$\underline{86.5}_{\pm 2.6}$	$77.0_{\pm 3.7}$	$\mathbf{90.5_{\pm 1.1}}$	$83.2_{\pm 5.8}$	$\mathbf{94.8_{\pm 2.7}}$	$\mathbf{92.2_{\pm 2.3}}$	$\mathbf{87.7_{\pm 1.6}}$	$\mathbf{90.2_{\pm 2.3}}$	$\mathbf{84.9_{\pm 1.6}}$

4 Experiments

4.1 Experimental Setup

Task domains. We evaluate GAVE on multi-turn agent tasks and on mathematical reasoning. For the agent tasks, we evaluate on ALFWorld (Shridhar et al., 2021), WebShop (Yao et al., 2022), and Search QA. ALFWorld is a text-based embodied environment for multi-step household tasks, where an agent receives a natural-language goal and completes it through sequential interaction with the environment. WebShop is an interactive online shopping environment where agents search, browse, and select products matching user instructions. Search QA evaluates search-augmented question answering in both single-hop and multi-hop settings. For mathematical reasoning, we train on DAPO-17K (Yu et al., 2025) and evaluate on AIME24 (Maxwell-Jia, 2024), AIME25 (Zhang & Math-AI Team, 2025), AMC23 (Qwen Team, 2024), MATH-500 (Lightman et al., 2024), and Olympiad-Bench (He et al., 2024).

Baselines. On ALFWorld and WebShop, we compare GAVE with GPT-4o (OpenAI, 2024), Gemini-2.5-Pro (Google DeepMind, 2025), ReAct (Yao et al., 2023), Reflexion (Shinn et al., 2023), PPO (MSE), trained with the standard mean-squared-error critic objective, PPO (BCE), trained with binary cross-entropy to predict terminal success, GRPO (Shao et al., 2024), and GiGPO (Feng et al., 2025), which estimates step-level advantages by grouping trajectories that share an environment state. On mathematical reasoning, we compare GAVE with PPO and GRPO.

Implementation details. We use Qwen3-4B and Qwen3-8B (Yang et al., 2025) for the agentic experiments, and Qwen3-4B-Base and Qwen3-8B-Base for mathematical reasoning. To ensure a controlled comparison, all RL methods within each domain use the same model initialization, data collection procedure, and optimization settings. Full training configurations, hyperparameters, and evaluation details are provided in Appendix B.

4.2 Main Results

Multi-turn agent tasks. Table 1 shows that GAVE achieves the best overall performance on ALFWorld and WebShop at both model sizes. On ALFWorld, GAVE improves overall success over GiGPO, the strongest baseline, by 13.3 and 5.2 percentage points for Qwen3-4B and Qwen3-8B, respectively. The gains are particularly pronounced on *Pick2* at both model sizes. The learning curves in Figure 1(b,c) complement these results: GAVE reaches high ALFWorld training rewards earlier and maintains higher evaluation success over most of training.

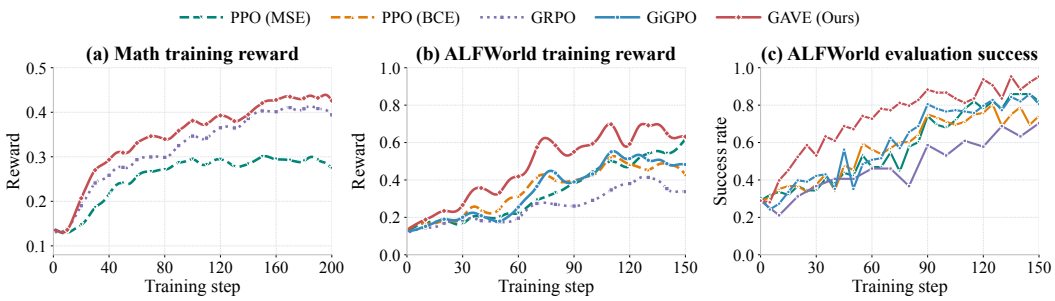


Figure 1: Training dynamics on mathematical reasoning and ALFWorld. (a) Math training reward. (b) ALFWorld training reward. (c) ALFWorld evaluation success. Reward curves in (a) and (b) are smoothed; (c) is plotted at the recorded validation steps.

On WebShop, GAVE achieves the highest scores and success rates. Compared with GiGPO, success rates improve by 8.9 and 8.1 percentage points at 4B and 8B, respectively, alongside score improvements of 5.5 and 4.9 points. Although PPO (BCE) substantially improves over PPO (MSE) on this benchmark, GAVE outperforms both variants. This comparison suggests that the gains extend beyond simply replacing the conventional critic loss with BCE.

The Search QA results in Appendix B.3 show that GAVE performs comparably to GiGPO on single-hop questions while outperforming it on all four multi-hop benchmarks. With Qwen3-4B, the average exact-match accuracy increases from 47.2% to 49.4%. These results support the effectiveness of GAVE across embodied, web-based, and search-based interaction.

Table 2: Accuracy (%) on mathematical reasoning benchmarks, reported as mean ± sample standard deviation over three seeds. Avg. is the unweighted mean across the five benchmarks. Within each model group, the highest mean is in bold and the second-highest mean is underlined.

Method	Mathematical Reasoning					
	AIME24	AIME25	AMC23	MATH-500	Olympiad-Bench	Avg.
Qwen3-4B-Base						
PPO	$12.6_{\pm 0.1}$	$9.4_{\pm 0.8}$	$46.7_{\pm 0.0}$	$76.8_{\pm 0.1}$	$38.5_{\pm 0.1}$	$36.8_{\pm 0.1}$
GRPO	$\underline{21.6}_{\pm 0.5}$	$\underline{18.3}_{\pm 1.4}$	$\underline{54.5}_{\pm 0.9}$	$\underline{82.2}_{\pm 0.2}$	$\underline{45.3}_{\pm 0.3}$	$\underline{44.4}_{\pm 0.3}$
GAVE (Ours)	$\mathbf{24.8}_{\pm \mathbf{0.4}}$	$\mathbf{19.2}_{\pm \mathbf{0.7}}$	$\mathbf{57.8}_{\pm \mathbf{0.0}}$	$\mathbf{83.1}_{\pm \mathbf{0.1}}$	$\mathbf{47.0}_{\pm \mathbf{0.2}}$	$\mathbf{46.4}_{\pm \mathbf{0.1}}$
Qwen3-8B-Base						
PPO	$\underline{24.4}_{\pm 1.3}$	$19.3_{\pm 1.3}$	$55.3_{\pm 1.7}$	$83.6_{\pm 0.1}$	$46.2_{\pm 0.4}$	$45.8_{\pm 0.4}$
GRPO	$\underline{24.4}_{\pm 1.4}$	$\mathbf{20.1}_{\pm \mathbf{0.6}}$	$\underline{59.7}_{\pm 1.1}$	$\underline{84.4}_{\pm 0.4}$	$\underline{48.1}_{\pm 0.1}$	$\underline{47.4}_{\pm 0.6}$
GAVE (Ours)	$\mathbf{29.4}_{\pm \mathbf{0.9}}$	$\underline{20.0}_{\pm 1.3}$	$\mathbf{61.8}_{\pm \mathbf{1.0}}$	$\mathbf{85.1}_{\pm \mathbf{0.3}}$	$\mathbf{50.1}_{\pm \mathbf{0.2}}$	$\mathbf{49.3}_{\pm \mathbf{0.2}}$

Mathematical reasoning. Table 2 presents the results on five mathematical reasoning benchmarks. GAVE achieves the best average accuracy at both model scales, reaching 46.4 on Qwen3-4B and 49.3 on Qwen3-8B, compared with 44.4 and 47.4 for GRPO, respectively. Across the individual benchmarks, GAVE ranks first in nine of the ten model–benchmark settings, showing that the improvement extends across both model scales and a diverse set of reasoning benchmarks. Notably, on AIME24 with Qwen3-8B, GAVE improves accuracy from 24.4 to 29.4, a gain of 5.0 percentage points over both PPO and GRPO.

Figure 1 further shows the corresponding training dynamics. GAVE achieves higher training reward than both PPO and GRPO and maintains this advantage through the later stages of training. Together, these results show that the gains of GAVE persist across model scales and are reflected in both downstream accuracy and training reward.

Table 3: Value estimation accuracy on held-out ALFWorld tasks, measured against Monte Carlo reference values. Methods marked with † share the same empirical prompt prior at evaluation.

	Overall Fit			Within-Prompt Fit	
Method	**MAE ↓**	**MSE ↓**	**EV ↑**	**MAE ↓**	**EV ↑**
Empirical Prior Only†	0.250	0.104	0.392	0.236	0.000
MSE Critic	0.177	0.065	0.616	0.160	0.543
BCE Critic	0.188	0.069	0.602	0.162	0.536
MSE Critic + Group Prior†	0.158	0.065	0.693	0.145	0.580
BCE Critic + Group Prior†	0.169	0.069	0.691	0.143	0.589
GAVE (Ours)†	**0.116**	**0.036**	**0.792**	**0.109**	**0.732**

4.3 Critic Value Estimation

To further examine how GAVE affects critic learning, we analyze the value estimator independently of policy optimization. Specifically, we examine (a) its overall and within-prompt prediction accuracy; and (b) the consistency of the gains across individual prefixes.

Value Approximation. Table 3 compares GAVE with MSE and BCE critics. To control for access to the empirical prompt prior, we also evaluate Empirical Prior Only and variants of both critics that receive the same prior as GAVE at evaluation. We report overall and within-prompt prediction accuracy, with the latter measured after centering predictions and Monte Carlo reference values separately within each prompt. Baseline definitions and evaluation metrics are detailed in Appendix C.

Both MSE and BCE critics outperform the Empirical Prior Only baseline, with further gains in MAE and EV when the group prior is incorporated. GAVE achieves the best overall accuracy, reducing MAE to 0.116 compared with 0.158 for MSE Critic + Group Prior. The gains extend to within-prompt prediction: GAVE achieves a Within-EV of 0.732, compared with 0.589 for BCE Critic + Group Prior. Together, these results show that directly learning the history-dependent component better captures within-prompt value variation than correcting conventional critics with the same empirical prior.

Prefix-Wise Error. While Table 3 summarizes prediction accuracy across prefixes, the paired comparisons in Fig. 2 provide a more fine-grained view of how the gains are distributed across individual evaluation prefixes. We compare GAVE with the MSE and BCE critics using both overall and within-prompt errors. Each point pairs the errors for the same evaluation prefix, so the comparison controls for differences in prefix difficulty, with points below the diagonal indicating lower error for GAVE.

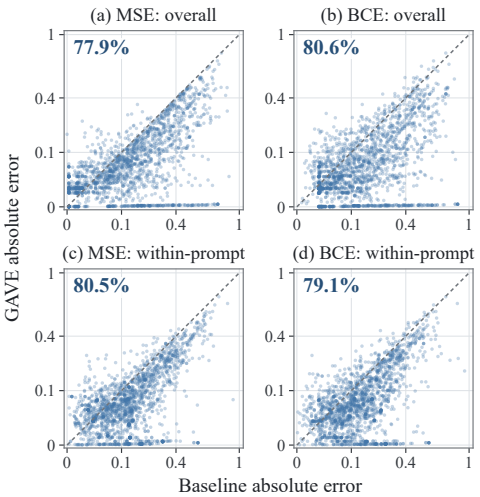


Figure 2: Prefix-level absolute errors. Percentages indicate the fraction of prefixes with lower error under GAVE. Both axes use square-root scaling, with ticks in original units.

GAVE achieves lower prediction errors on 77.9% and 80.6% of prefixes than the MSE and BCE critics, respectively. After within-prompt centering, the corresponding proportions are 80.5% and 79.1%. These paired results complement the aggregate metrics in Table 3, which also depend on the magnitude of individual prediction errors, by showing how frequently GAVE improves on the same prefixes. Overall, the gains are broadly distributed across the evaluation set rather than being driven by a small subset of prefixes, and this pattern remains after removing prompt-level offsets to focus on value variation within the same prompt.

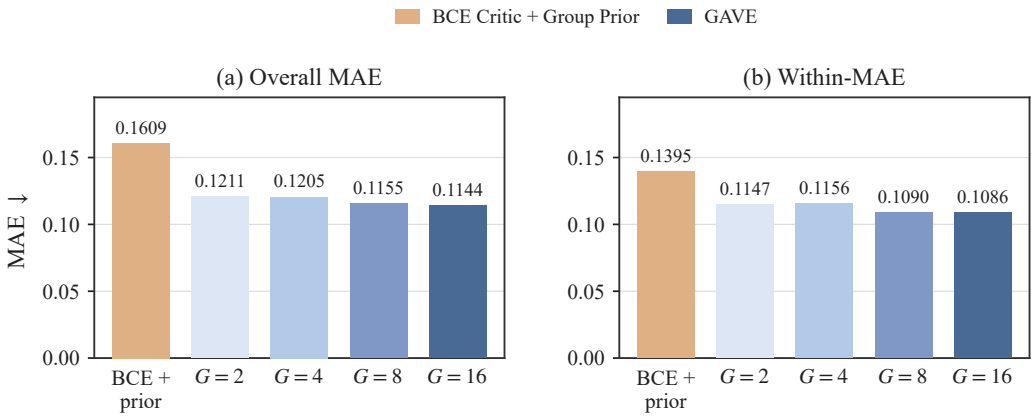


Figure 3: Effect of training group size on value prediction. (a) Overall MAE and (b) Within-MAE for GAVE with different training group sizes and the BCE CRITIC + GROUP PRIOR baseline. All settings use the same training rollouts and evaluation data. Lower is better.

4.4 ABLATION STUDY

We study how the training group size affects value prediction in GAVE. Using the frozen-policy ALFWorld setting from our critic evaluation, we vary G over $\{2, 4, 8, 16\}$. For each training prompt, we randomly partition the same 16 rollouts into groups of size G. This changes the group statistics used in the critic objective while keeping the training trajectories fixed. We use the same critic initialization and optimization budget across settings. At evaluation, all variants share the same prefixes, Monte Carlo reference values, and empirical prompt priors. The BCE CRITIC + GROUP PRIOR baseline is trained on the same data. We also train BCE CRITIC + GROUP PRIOR on the same data as a reference.

Figure 3 shows that GAVE achieves lower MAE and Within-MAE than BCE CRITIC + GROUP PRIOR across all tested group sizes, from $G = 2$ to $G = 16$. This advantage is present even at $G = 2$, showing that the group-conditional objective remains effective when training rollouts are grouped in pairs. The lower Within-MAE further indicates that GAVE better captures value variation among prefixes from the same prompt.

Increasing the training group size generally improves both metrics. The accompanying reduction in Within-MAE suggests that larger groups also benefit the critic's estimation of history-dependent value differences. Most of the observed improvement is obtained by $G = 8$, while doubling the group size to $G = 16$ yields only modest additional gains.

5 RELATED WORK

Credit assignment in LLM post-training. PPO uses a learned value function and Generalized Advantage Estimation to construct advantages at intermediate states (Schulman et al., 2017; 2016). RLOO and GRPO replace the learned critic with advantages computed from the rewards of multiple responses to the same prompt (Ahmadian et al., 2024; Shao et al., 2024). GiGPO extends group-based credit assignment to interactive tasks by comparing actions taken from the same environment state across trajectories (Feng et al., 2025). VinePPO estimates intermediate values directly through Monte Carlo continuations, replacing learned value predictions with sampling-based estimates (Kazemnejad et al., 2025). Process reward models provide another source of fine-grained supervision by assigning feedback to intermediate reasoning steps (Lightman et al., 2024; Wang et al., 2024). GAVE uses grouped terminal outcomes to train a critic for intermediate-state value estimation, without additional branching rollouts or changes to the reward signal.

Learning value functions. Recent work has revisited how to train reliable and efficient critics for LLM reinforcement learning. VC-PPO introduces value pretraining and separate GAE param-

eters for critic targets and policy advantages (Yuan et al., 2025). VAPO builds on these designs with length-adaptive GAE, accounting for the varying lengths of reasoning trajectories (Yue et al., 2025). Beyond initialization and return estimation, Stop Regressing studies the loss used to fit value functions, showing that categorical cross-entropy can improve performance and scalability in deep RL (Farebrother et al., 2024). Other work examines how learned values can be used alongside empirical baselines when critic reliability varies during training. EVPO uses explained variance to switch between the critic and a mean-return baseline (Pan et al., 2026), while TETHER adaptively interpolates between a group baseline and token-level value predictions (Venkatraman et al., 2026). GAVE instead uses grouped outcomes to define what the critic learns: its group-conditional objective targets the history-dependent component of success log-odds, which is combined with an empirical prompt prior to reconstruct on-policy state values.

6 Conclusion

We introduced GAVE, a training formulation that uses grouped rollouts to improve critic learning in LLM reinforcement learning. For binary terminal rewards, a Bayesian decomposition separates state-value log-odds into a prompt-level prior term and history-dependent evidence. GAVE estimates the prior from grouped outcomes and learns the evidence through a group-conditional objective. Our analysis characterizes how errors in these two components affect the reconstructed value. With exact history evidence, the expected absolute reconstruction error under the on-policy state distribution is bounded by the absolute error in the prompt prior. The reconstructed values are used for advantage estimation within the existing PPO pipeline. Prior estimation reuses the outcomes of grouped training rollouts, and the policy objective remains unchanged.

Our critic evaluation on ALFWorld shows that GAVE improves both overall prediction accuracy and the estimation of value variation within a prompt. These improvements persist against MSE and BCE critics given the same empirical prompt prior at evaluation, supporting the benefit of learning the history-dependent component directly. Paired comparisons at individual prefixes further show that the gains are broadly distributed across the evaluation states. Policy optimization experiments provide complementary evidence, with improved performance on multi-turn agent tasks and mathematical reasoning benchmarks. Together, these findings highlight the value of using rollout group structure to determine what can be estimated from empirical statistics and what the critic should learn.

AI USE STATEMENT

During manuscript preparation, we used generative AI tools as editorial and programming assistants. Their use was limited to improving prose, suggesting organizational revisions, checking LaTeX, and helping diagnose presentation or implementation issues. The tools did not determine the research problem, method, theoretical claims, experimental design, or interpretation of the results. The authors independently made and verified those decisions, reviewed every AI-assisted change, and tested any retained code. No data or experimental result was generated by an AI system, and the authors accept responsibility for the accuracy, originality, and integrity of the submitted work.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to Basics: Revisiting REINFORCE-Style Optimization for Learning from Human Feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.662. URL https://aclanthology.org/2024.acl-long.662/.

DeepSeek-AI. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*, 2025.

Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taiga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, Aviral Kumar, and Rishabh Agarwal. Stop Regressing: Training Value Functions via Classification for Scalable Deep RL. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=dVpFKfqF3R.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-Group Policy Optimization for LLM Agent Training. In *Advances in Neural Information Processing Systems*, pp. 46375–46408, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/420c9f777c0b4f78d515e53cf74d58b2-Paper-Conference.pdf.

Google DeepMind. Gemini 2.5 Pro Model Card, 2025. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-2-5-Pro-Model-Card.pdf.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A Challenging Benchmark for Promoting AGI with Olympiad-Level Bilingual Multimodal Scientific Problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625. International Committee on Computational Linguistics, 2020. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning. *arXiv preprint arXiv:2503.09516*, 2025. URL https://arxiv.org/abs/2503.09516.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining Credit Assignment in RL Training of LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29557–29590. PMLR, 2025. URL https://proceedings.mlr.press/v267/kazemnejad25a.html.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

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 *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

Maxwell-Jia. AIME 2024 Dataset, 2024. URL https://huggingface.co/datasets/Maxwell-Jia/AIME_2024.

OpenAI. GPT-4o System Card, 2024. URL https://openai.com/index/gpt-4o-system-card/.

OpenAI, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. OpenAI o1 System Card. *arXiv preprint arXiv:2412.16720*, 2024.

Chengjun Pan, Shichun Liu, Jiahang Lin, Dingwei Zhu, Jiazheng Zhang, Shihan Dou, Songyang Gao, Zhenhua Han, Binghai Wang, Rui Zheng, Xuanjing Huang, Tao Gui, and Yansong Feng. EVPO: Explained Variance Policy Optimization for Adaptive Critic Utilization in LLM Post-Training. *arXiv preprint arXiv:2604.19485*, 2026. URL https://arxiv.org/abs/2604.19485.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and Narrowing the Compositionality Gap in Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.378. URL https://aclanthology.org/2023.findings-emnlp.378/.

Qwen Team. AMC23 Benchmark Dataset, 2024. URL https://github.com/QwenLM/Qwen2.5-Math/blob/main/evaluation/data/amc23/test.jsonl.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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. *arXiv preprint arXiv:2402.03300*, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language Agents with Verbal Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.03768.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. ZeroSearch: Incentivize the Search Capability of LLMs without Searching. *arXiv preprint arXiv:2505.04588*, 2025. URL https://arxiv.org/abs/2505.04588.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Siddarth Venkatraman, Matthieu Dinot, and Laurence Aitchison. Le Critique: Privileged Value Functions for LLM Reinforcement Learning. *arXiv preprint arXiv:2608.16739*, 2026. URL https://arxiv.org/abs/2608.16739.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

Ziliang Wang, Xuhui Zheng, Kang An, Cijun Ouyang, Jialu Cai, Yuhang Wang, and Yichao Wu. StepSearch: Igniting LLMs Search Ability via Step-Wise Proximal Policy Optimization. *arXiv preprint arXiv:2505.15107*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025.

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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2207.01206.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. *arXiv preprint arXiv:2503.14476*, 2025.

Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. What's Behind PPO's Collapse in Long-CoT? Value Optimization Holds the Secret. *arXiv preprint arXiv:2503.01491*, 2025. URL https://arxiv.org/abs/2503.01491.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, Xiangyu Yu, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Ru Zhang, Xin Liu, Mingxuan Wang, Yonghui Wu, and Lin Yan. VAPO: Efficient and Reliable Reinforcement Learning for Advanced Reasoning Tasks. *arXiv preprint arXiv:2504.05118*, 2025. URL https://arxiv.org/abs/2504.05118.

Yifan Zhang and Math-AI Team. American Invitational Mathematics Examination (AIME) 2025, 2025. URL https://huggingface.co/datasets/math-ai/aime25.

A PROOFS

A.1 PROOF OF PROPOSITION 1

Proof. For $0 < k < G$, exchangeability of the rollouts conditional on x gives

$$P^\pi(R_i = 1 \mid x, K = k) = \frac{k}{G}, \qquad P^\pi(R_i = 0 \mid x, K = k) = \frac{G-k}{G}.$$

Write $K_{-i} = \sum_{j \neq i} R_j$. Given $R_i = r$, the condition $K = k$ is equivalent to $K_{-i} = k - r$. Since K_{-i} depends only on the other rollouts, conditional independence gives

$$P^\pi(y^{(i)}_{<t} \mid x, R_i = r, K = k) = P^\pi(y^{(i)}_{<t} \mid x, R_i = r), \qquad r \in \{0, 1\}.$$

Bayes' rule then yields

$$\begin{aligned}\frac{P^\pi(R_i = 1 \mid s_{i,t}, K = k)}{P^\pi(R_i = 0 \mid s_{i,t}, K = k)} &= \frac{P^\pi(y^{(i)}_{<t} \mid x, R_i = 1)}{P^\pi(y^{(i)}_{<t} \mid x, R_i = 0)} \frac{k}{G-k} \\ &= \exp(\Delta^\pi(s_{i,t}))\, \frac{k}{G-k},\end{aligned}$$

where the second equality follows from the definition of Δ^π. Rearranging gives

$$P^\pi(R_i = 1 \mid s_{i,t}, K = k) = \sigma\left(\log \frac{k}{G-k} + \Delta^\pi(s_{i,t})\right).$$

When $k = 0$, all outcomes are zero; when $k = G$, all outcomes are one. The conditional success probabilities are therefore 0 and 1, respectively, as specified by the endpoint convention. □

A.2 PROOF OF PROPOSITION 2

Proof. The sigmoid satisfies $\sigma'(z) = \sigma(z)(1 - \sigma(z)) \leq 1/4$. Since $g(\widetilde{p}, \delta) = \sigma(\operatorname{logit} \widetilde{p} + \delta)$ for $0 < \widetilde{p} < 1$, we have

$$\big|g(\widetilde{p}, f_\phi(s)) - g(\widetilde{p}, \Delta^\pi(s))\big| \leq \frac{1}{4}\big|f_\phi(s) - \Delta^\pi(s)\big|.$$

To bound the error from the prior, let μ_r denote the density of $P^\pi(\cdot \mid x, R = r)$ with respect to a common measure ν, and write

$$d_q(s) = q\mu_1(s) + (1-q)\mu_0(s).$$

Thus, d_p is the density of $P^\pi(\cdot \mid x)$. The definition of Δ^π gives

$$g(q, \Delta^\pi(s)) = \frac{q\mu_1(s)}{d_q(s)}, \qquad 0 < q < 1.$$

Taking the difference at $q = \widetilde{p}$ and $q = p$ yields

$$g(\widetilde{p}, \Delta^\pi(s)) - g(p, \Delta^\pi(s)) = \frac{(\widetilde{p} - p)\mu_1(s)\mu_0(s)}{d_{\widetilde{p}}(s) d_p(s)}.$$

Integrating against d_p therefore gives

$$\begin{aligned}\mathbb{E}_s\big|g(\widetilde{p}, \Delta^\pi(s)) - g(p, \Delta^\pi(s))\big| &= |\widetilde{p} - p| \int \frac{\mu_1(s)\mu_0(s)}{d_{\widetilde{p}}(s)}\, d\nu(s) \\ &\leq |\widetilde{p} - p| \int \big[(1-\widetilde{p})\mu_1(s) + \widetilde{p}\mu_0(s)\big]\, d\nu(s) \\ &= |\widetilde{p} - p|,\end{aligned}$$

where $\mathbb{E}_s$ denotes expectation under $P^\pi(\cdot \mid x)$. The inequality follows from

$$\big[(1-q)a + qb\big]\big[qa + (1-q)b\big] - ab = q(1-q)(a-b)^2 \geq 0,$$

with $a = \mu_1(s)$, $b = \mu_0(s)$, and $q = \widetilde{p}$. The final equality uses $\int \mu_0 \, d\nu = \int \mu_1 \, d\nu = 1$.

Combining the two bounds with $V^\pi(s) = g(p, \Delta^\pi(s))$ gives

$$\begin{aligned}\mathbb{E}_s\big|g(\widetilde{p}, f_\phi(s)) - V^\pi(s)\big| &\le \mathbb{E}_s\big|g(\widetilde{p}, f_\phi(s)) - g(\widetilde{p}, \Delta^\pi(s))\big| \\ &\quad + \mathbb{E}_s\big|g(\widetilde{p}, \Delta^\pi(s)) - g(p, \Delta^\pi(s))\big| \\ &\le \frac{1}{4}\mathbb{E}_s\big|f_\phi(s) - \Delta^\pi(s)\big| + |\widetilde{p} - p|.\end{aligned}$$

At $\widetilde{p} = 0$ and $\widetilde{p} = 1$, the prediction is identically zero and one, respectively. Since $\mathbb{E}_s[V^\pi(s)] = p$, the expected error equals $|\widetilde{p} - p|$ in either case. □

A.3 Proof of Corollary 1

Proof. The smoothed prior $\widetilde{p}_m = (\sum_{j=1}^m R_j + \alpha)/(m + 2\alpha)$ is independent of the evaluation state s conditional on x. Conditioning on $\widetilde{p}_m$ leaves the state distribution unchanged, so Proposition 2 gives

$$\mathbb{E}\left[\big|\widehat{V}_m(s) - V^\pi(s)\big| \,\middle|\, \widetilde{p}_m\right] \le |\widetilde{p}_m - p| + \frac{1}{4}\varepsilon_\phi.$$

Averaging over $\widetilde{p}_m$ yields

$$\mathbb{E}\big|\widehat{V}_m(s) - V^\pi(s)\big| \le \mathbb{E}|\widetilde{p}_m - p| + \frac{1}{4}\varepsilon_\phi.$$

The sibling outcomes are independent Bernoulli(p) variables. Hence, $\widetilde{p}_m$ has mean $(mp + \alpha)/(m + 2\alpha)$ and variance $mp(1-p)/(m+2\alpha)^2$. Its bias is therefore $\alpha(1-2p)/(m+2\alpha)$. Using Cauchy–Schwarz and the bias–variance decomposition gives

$$\mathbb{E}|\widetilde{p}_m - p| \le \sqrt{\mathbb{E}\big[(\widetilde{p}_m - p)^2\big]} = \frac{\sqrt{mp(1-p) + \alpha^2(1-2p)^2}}{m + 2\alpha}.$$

Substituting this bound proves the corollary. □

B Experimental and Implementation Details

B.1 Training Procedure

Algorithm 1 summarizes the full GAVE training procedure. GAVE retains the standard PPO rollout and policy-update pipeline, while modifying the critic training objective and the construction of state values. For each prompt, the grouped rollout outcomes are used to form the group-conditional critic objective and a leave-one-out estimate of the prompt prior. The latter depends only on group-level reward statistics and is combined with the critic output to reconstruct the state values used by GAE. These additions require no changes to the remaining PPO pipeline.

B.2 ALFWorld and WebShop

Hyperparameters. ALFWorld and WebShop use the same training configuration unless otherwise specified. The maximum prompt length is 4096 tokens, and the maximum response length is 512 tokens. The actor learning rate is set to 10^{-6}, while the critic learning rate is 10^{-5} for methods with a learned critic. We use a rule-based binary reward, assigning a reward of 1 for successful task completion and 0 for failure. For methods using grouped rollouts, we use a group size of 8 and sample 16 groups per rollout, resulting in 128 environments in total. The rollout temperature is set to 1.0, while the validation temperature is set to 0.4. The KL-divergence coefficient is set to 0.01. ALFWorld allows up to 50 environment steps per episode with a mini-batch size of 256, while WebShop uses at most 15 steps with a mini-batch size of 64. For GAVE, we set the smoothing coefficient in the leave-one-out prompt prior to $\alpha = 0.5$ and use $\gamma = 1$ and $\lambda = 1$ for GAE.

Computing details. For both Qwen3-4B and Qwen3-8B, all ALFWorld and WebShop experiments are trained for 150 iterations using $8\times$ H200 GPUs.

Algorithm 1 Group-Anchored Value Estimation (GAVE) with PPO

Require: Policy π_θ, evidence critic f_ϕ, prompt distribution $\mathcal{D}$, group size G, smoothing coefficient $\alpha > 0$, GAE parameter λ, PPO clipping parameter ϵ

for each training iteration **do**
  $\theta_{\text{old}} \leftarrow \theta$, $\phi_{\text{old}} \leftarrow \phi$
  Sample a batch of prompts $\mathcal{X}$ from $\mathcal{D}$
  for each prompt $x \in \mathcal{X}$ **do**
    Collect G independent rollouts $\{y^{(i)}, R_i\}_{i=1}^{G}$ under $\pi_{\theta_{\text{old}}}(\cdot \mid x)$
    $K \leftarrow \sum_{i=1}^{G} R_i$
    $\widetilde{p}_{-i}(x) \leftarrow \frac{K - R_i + \alpha}{G - 1 + 2\alpha}$ for all i
    $\widehat{V}_{i,t} \leftarrow \sigma\big(\operatorname{logit} \widetilde{p}_{-i}(x) + f_{\phi_{\text{old}}}(s_{i,t})\big)$ for all i, t
    Compute advantages $\widehat{A}_{i,t}$ from $\widehat{V}_{i,t}$ using Eq. (3)
  end for
  Update θ by maximizing $\mathcal{J}_{\text{PPO}}(\theta)$ in Eq. (2)
  Update ϕ by minimizing $\mathcal{L}_{\text{critic}}(\phi)$ in Eq. (9)
end for

B.3 Search-Augmented QA

Datasets. We train on Natural Questions (NQ) (Kwiatkowski et al., 2019) and HotpotQA (Yang et al., 2018) and evaluate on seven search-augmented QA benchmarks. The single-hop benchmarks include NQ, TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023), while the multi-hop benchmarks include HotpotQA, 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023). NQ and HotpotQA are used for in-domain evaluation, whereas TriviaQA, PopQA, 2WikiMultiHopQA, MuSiQue, and Bamboogle are held out during training and used for out-of-domain evaluation.

Hyperparameters. The maximum prompt length is 4096 tokens, and the maximum response length is 512 tokens. Each episode allows at most 4 search turns. The actor learning rate is set to 10^{-6}, and the critic learning rate is set to 10^{-5} for GAVE. We use a binary reward of 1 for a correct final answer and 0 otherwise. We use a prompt batch size of 256, and the group size is set to 5. The rollout and evaluation temperatures are set to 1.0 and 0.4, respectively. The mini-batch size is 512, and the KL-divergence coefficient is set to 0.001. For GAVE, we set the smoothing coefficient to $\alpha = 0.5$ and use $\gamma = 1$ and $\lambda = 1$ for GAE.

Performance. Table 4 reports the results on search-augmented QA. For the matched Qwen3-4B comparison, GAVE performs comparably to GiGPO on the single-hop benchmarks, while achieving consistently stronger results on the multi-hop benchmarks. In particular, GAVE outperforms GiGPO on HotpotQA, 2WikiMultiHopQA, MuSiQue, and Bamboogle, leading to a higher overall average. The stronger performance on multi-hop QA is consistent with settings in which successful search requires credit assignment across a sequence of intermediate retrieval decisions rather than a single retrieval step.

B.4 Mathematical Reasoning

Dataset and prompt format. We train on the full DAPO-Math-17K training set, which contains 17,917 mathematical reasoning problems. Each problem is paired with a ground-truth final answer for rule-based verification. For both training and evaluation, we use the following prompt template:

```
<|im_start|>user
{problem}

Please reason step by step, and put your final answer within
\boxed{}.<|im_end|>
<|im_start|>assistant
```

Table 4: Exact Match (EM) performance on seven search-augmented QA benchmarks. GAVE and GiGPO are trained on NQ and HotpotQA using Qwen3-4B. † and ⋆ denote in-domain and out-of-domain datasets, respectively.

Method	Type	Single-Hop QA			Multi-Hop QA				Avg.
		NQ†	TriviaQA⋆	PopQA⋆	HotpotQA†	2Wiki⋆	MuSiQue⋆	Bamboogle⋆	
Prompting									
GPT-4o (OpenAI, 2024)	Closed-source	-	-	-	-	-	-	-	-
Qwen3-32B (Yang et al., 2025)	Open-source	13.56	41.32	14.28	18.24	25.77	3.98	12.32	21.58
RL Training (Based on `Qwen2.5-7B-Instruct`)									
R1-Instruct	Open-source	21.0	44.9	17.1	20.8	27.5	6.0	19.2	22.4
Search-R1 (Jin et al., 2025)	Open-source	39.3	61.0	39.7	37.0	40.1	14.6	36.8	38.5
ZeroSearch (Sun et al., 2025)	Open-source	43.6	61.8	**51.5**	34.6	35.2	18.4	27.8	39.1
StepSearch (Wang et al., 2025)	Open-source	-	-	-	38.6	36.6	**22.6**	40.0	-
RL Training (Based on `Qwen3-4B`)									
GiGPO (Feng et al., 2025)	Open-source	47.4	64.9	**48.5**	42.3	39.5	17.6	70.2	47.2
GAVE (Ours)	Open-source	**48.7**	**65.2**	47.7	**45.8**	**44.0**	**20.2**	**74.2**	**49.4**

Reward. We use a rule-based correctness reward. The final answer is extracted from the generated response and verified against the ground-truth answer using Math-Verify. A correct response receives a reward of 1, and an incorrect response receives a reward of 0.

Training configuration. We initialize the policies from Qwen3-4B-Base and Qwen3-8B-Base. For each model scale, PPO, GRPO, and GAVE use the same rollout batch size, generation settings, and actor optimization hyperparameters unless otherwise specified. At each rollout step, we sample 128 prompts and generate eight responses per prompt, yielding 1,024 trajectories. The maximum training response length is 8,192 tokens, and rollouts are sampled with temperature 1.0 and top-$p = 1.0$.

The actor learning rate is 5×10^{-6} and follows a cosine schedule with 15 warmup steps and a minimum learning-rate ratio of 0.1. We use the token-level clipped policy-gradient objective with $\epsilon_{\text{low}} = 0.20$ and $\epsilon_{\text{high}} = 0.28$, and do not use an entropy bonus or KL-divergence regularization. PPO (MSE) and GAVE estimate advantages using GAE with $\gamma = 1$ and $\lambda = 0.995$. GRPO computes group-relative advantages from the eight responses sampled for each prompt. For all methods and seeds, we report the checkpoint at step 200. Table 5 summarizes the remaining training and evaluation settings.

Evaluation benchmarks. We evaluate mathematical reasoning on the following five benchmarks.

- **AIME24 and AIME25.** AIME24 (Maxwell-Jia, 2024) and AIME25 (Zhang & Math-AI Team, 2025) comprise questions from the 2024 and 2025 editions of the American Invitational Mathematics Examination, respectively. Each benchmark contains 30 problems drawn from AIME I and AIME II. The problems cover algebra, geometry, number theory, and combinatorics, with integer answers ranging from 0 to 999.
- **AMC23.** AMC23 (Qwen Team, 2024) contains 40 questions selected from the 2023 AMC 12A and AMC 12B examinations. It covers secondary-school mathematics, including arithmetic, algebra, geometry, number theory, combinatorics, and probability.
- **MATH-500.** MATH-500 (Lightman et al., 2024) consists of 500 problems selected from the test split of MATH. It spans seven subjects: prealgebra, algebra, intermediate algebra, number theory, counting and probability, geometry, and precalculus. The problems have difficulty labels from 1 to 5 and are accompanied by reference solutions and final answers.
- **OlympiadBench.** OlympiadBench (He et al., 2024) contains mathematics and physics problems drawn from Olympiad competitions and Chinese college entrance examinations. The full benchmark includes questions in English and Chinese, with both text-only and multimodal inputs. It provides expert-annotated solutions together with annotations of subject areas and answer types.

Evaluation protocol. For each of three independent training runs (seeds 42, 43, and 44), we sample 16 responses per problem. We use a maximum response length of 16,384 tokens, a temperature of 0.6, and top-$p = 0.95$. Table 2 reports the mean and sample standard deviation across the three runs.

Table 5: Hyperparameters for mathematical-reasoning training and evaluation.

Group	Parameter	Value
Data	Training examples	17,917
	Rollout prompt batch size	128
	Maximum prompt length	1,024
Actor	Learning rate	5×10^{-6}
	Mini-batch size	128
	Clipping parameters $(\epsilon_{\text{low}}, \epsilon_{\text{high}})$	$0.20/0.28$
	Entropy coefficient	0
	KL regularization	None
	Loss aggregation	token mean
	Learning-rate schedule	cosine
	Warmup steps	15
	Minimum learning-rate ratio	0.1
Critic (PPO/GAVE)	Learning rate	1×10^{-5}
	Mini-batch size	128
	Value clipping threshold	0.5
Rollout	Responses per prompt (G)	8
	Maximum response length	8,192
	Temperature / top-p	$1.0/1.0$
Evaluation	Reported checkpoint	step 200
	Maximum response length	16,384
	Samples per problem	16
	Temperature / top-p	$0.6/0.95$

For the overall score, we average the five benchmark scores within each run before computing the mean and sample standard deviation across runs.

Compute. Each mathematical-reasoning run uses eight NVIDIA H200 GPUs.

C Critic Evaluation

C.1 Experimental Setup

We evaluate the critics on ALFWorld under a fixed policy. Throughout critic training and evaluation, we use the policy checkpoint at training step 50, denoted by π. This allows all methods to be compared against the same value function V^{π}.

Critic training. All learned critics use the Qwen3-4B backbone and are trained on the same dataset of grouped rollouts collected under π, with eight trajectories per prompt. We use the same optimization settings and training budget across methods. Table 6 summarizes the training configuration.

Evaluation data. We evaluate critic predictions at turn-level prefixes from trajectories generated by π on 32 held-out ALFWorld tasks. All critic variants are evaluated on the same set of prefixes.

For methods that use an empirical prompt prior, we collect eight additional rollouts from the initial state of each held-out task. These rollouts are sampled independently of both the evaluation trajectories and the continuations used to construct the reference values below. The same prior rollouts are used for GAVE and all baselines that receive prompt-level information.

Monte Carlo reference values. For each evaluation prefix s, we sample $M = 64$ independent continuations under π and compute the reference value

$$\widehat{V}_{\text{MC}}(s) = \frac{1}{M} \sum_{m=1}^{M} R^{(m)}, \tag{16}$$

where $R^{(m)} \in \{0, 1\}$ is the terminal reward of the m-th continuation from s. We use these continuations only for evaluation and compare all methods on the same prefixes and reference values.

C.2 Critic Variants

We compare GAVE with MSE and BCE critics, as well as baselines that use the same empirical prompt prior at evaluation.

MSE and BCE critics. We train two baseline critics using the terminal reward of each trajectory as the target for its intermediate states. The MSE Critic minimizes mean squared error, whereas the BCE Critic uses binary cross-entropy. The BCE baseline controls for the use of binary cross-entropy in GAVE.

Empirical prompt prior. For each held-out prompt x, we estimate the prompt prior from the eight independent rollouts described above:

$$\widehat{p}(x) = \frac{1}{G} \sum_{j=1}^{G} R^{(j)}, \qquad G = 8, \tag{17}$$

where $R^{(j)}$ is the terminal reward of the j-th rollout from x. GAVE and all baselines with prior access use the same estimate for each prompt. The Empirical Prior Only baseline assigns $\widehat{V}(s) = \widehat{p}(x)$ to every prefix associated with x. This baseline measures the predictive accuracy available from prompt-level information alone.

Critics with a group prior. We additionally evaluate MSE Critic + Group Prior and BCE Critic + Group Prior. Both variants correct the predictions of the corresponding trained critic at evaluation time. Let $\widehat{V}(s)$ denote its value prediction and s_0 the initial state for prompt x. We compute the predicted change in log-odds and combine it with the empirical prompt prior:

$$\begin{aligned} \widehat{\Delta}(s) &\coloneqq \operatorname{logit} \bar{V}(s) - \operatorname{logit} \bar{V}(s_0), \\ \widehat{V}_{\mathrm{GP}}(s) &= g\Big(\widehat{p}(x), \widehat{\Delta}(s)\Big). \end{aligned} \tag{18}$$

Here, $\bar{V}(s) = \mathrm{clip}(\widehat{V}(s), \epsilon_{\mathrm{eval}}, 1 - \epsilon_{\mathrm{eval}})$ ensures finite log-odds, and g is defined in Eq. (12). These variants control for access to the empirical prompt prior at evaluation.

GAVE. GAVE trains the evidence critic f_ϕ using the group-conditional objective in Eq. (9). At evaluation, we combine its output with the same empirical prompt prior:

$$\widehat{V}_{\mathrm{GAVE}}(s) = g(\widehat{p}(x), f_\phi(s)). \tag{19}$$

This comparison isolates the benefit of directly learning the history-dependent component, beyond simply correcting standard MSE and BCE critics with the same empirical prompt prior.

C.3 Evaluation Metrics

We evaluate each critic prediction $\widehat{V}(s)$ against the Monte Carlo reference $\widehat{V}_{\mathrm{MC}}(s)$.

Overall accuracy. We report mean absolute error (MAE), mean squared error (MSE), and explained variance (EV). MAE and MSE quantify prediction error, while EV measures the reduction in residual variance relative to the variance of the reference values:

$$\mathrm{EV} = 1 - \frac{\mathrm{Var}_s\Big[\widehat{V}_{\mathrm{MC}}(s) - \widehat{V}(s)\Big]}{\mathrm{Var}_s\Big[\widehat{V}_{\mathrm{MC}}(s)\Big]}. \tag{20}$$

All metrics are computed over the evaluation prefixes, with equal weight assigned to each prefix. Lower MAE and MSE and higher EV indicate better prediction quality.

Table 6: Shared configuration for critic training under the fixed policy. All learned critics use the same model backbone, rollout data, and optimization settings.

Group	Parameter	Value
Model	Backbone	Qwen3-4B
	Training mode	Offline full-parameter fine-tuning
Rollout data	Benchmark	ALFWorld
	Data-collection policy	Step-50 checkpoint (frozen)
	Training tasks	1024
	Rollouts per prompt (G)	8
	Temperature / top-p	1.0 / 1.0
Optimization	Optimizer	AdamW
	Learning rate	1×10^{-5}
	Batch size	128
	Optimizer updates	150
	Learning-rate schedule	Constant
	Warmup updates	0
	Weight decay	0.01
	Maximum gradient norm	1.0

Within-prompt accuracy. To assess how well a critic captures value variation among prefixes from the same prompt, we center predictions and reference values within each prompt:

$$\begin{aligned} \widehat{V}^{\,c}(s) &= \widehat{V}(s) - \mu(x), \\ \widehat{V}^{c}_{\mathrm{MC}}(s) &= \widehat{V}_{\mathrm{MC}}(s) - \mu_{\mathrm{MC}}(x). \end{aligned} \tag{21}$$

Here, $\mu(x)$ and $\mu_{\mathrm{MC}}(x)$ are the respective empirical means of the predictions and reference values over evaluation prefixes associated with prompt x. We report Within-MAE and Within-EV by applying the same metrics to the centered predictions and reference values. The centered values are used exclusively to compute the within-prompt metrics.

D Extension to Finite Discrete Rewards

We extend GAVE to terminal rewards $R \in \{r_0, \ldots, r_{C-1}\}$, where $C \geq 2$. The extension uses a single critic network with a vector-valued output head to model the relative evidence for each reward outcome. The scalar value is then recovered as the expected terminal reward.

D.1 Group-Conditional Reward Distribution

For a fixed policy π, let $p^{\pi}_{c}(x) \coloneqq P^{\pi}(R = r_c \mid x)$ denote the prompt-level probability of outcome r_c. Taking r_0 as the reference outcome, define

$$\Delta^{\pi}_{c}(s_t) \coloneqq \log \frac{P^{\pi}(y_{<t} \mid x, R = r_c)}{P^{\pi}(y_{<t} \mid x, R = r_0)}, \qquad c = 0, \ldots, C-1. \tag{22}$$

Thus, $\Delta^{\pi}_{0} \equiv 0$. We assume positive prompt probabilities and positive class-conditional history likelihoods for the outcomes and histories considered. Bayes' rule gives the categorical counterpart of Eq. (7):

$$P^{\pi}(R = r_c \mid s_t) = \frac{p^{\pi}_{c}(x) \exp\big(\Delta^{\pi}_{c}(s_t)\big)}{\sum_{\ell=0}^{C-1} p^{\pi}_{\ell}(x) \exp\big(\Delta^{\pi}_{\ell}(s_t)\big)}. \tag{23}$$

The prompt probabilities determine the initial distribution over rewards, while the history evidence updates their relative likelihoods.

Now consider $G \geq 2$ independent rollouts from the same prompt under π. Let $\mathbf{K} = (K_0, \ldots, K_{C-1})$ be their reward counts, where $K_c = \sum_{j=1}^{G} \mathbf{1}\{R_j = r_c\}$. Conditioning on these counts removes the prompt probabilities from the prediction problem.

Proposition 3 (Group-conditional reward distribution)**.** *For any count vector* $\mathbf{k} = (k_0, \ldots, k_{C-1})$ *with* $\sum_{c=0}^{C-1} k_c = G$*,*

$$P^\pi(R_i = r_c \mid s_{i,t}, \mathbf{K} = \mathbf{k}) = \frac{k_c \exp\big(\Delta_c^\pi(s_{i,t})\big)}{\sum_{\ell=0}^{C-1} k_\ell \exp\big(\Delta_\ell^\pi(s_{i,t})\big)}. \tag{24}$$

Proof. Exchangeability gives $P^\pi(R_i = r_c \mid x, \mathbf{K} = \mathbf{k}) = k_c/G$. For $k_c > 0$, once $R_i = r_c$ is fixed, the remaining count constraint concerns only the sibling rollouts. Their independence implies

$$P^\pi(y_{<t}^{(i)} \mid x, R_i = r_c, \mathbf{K} = \mathbf{k}) = P^\pi(y_{<t}^{(i)} \mid x, R_i = r_c).$$

Applying Bayes' rule therefore yields

$$P^\pi(R_i = r_c \mid s_{i,t}, \mathbf{K} = \mathbf{k}) = \frac{k_c P^\pi(y_{<t}^{(i)} \mid x, R_i = r_c)}{\sum_{\ell=0}^{C-1} k_\ell P^\pi(y_{<t}^{(i)} \mid x, R_i = r_\ell)}.$$

Dividing the numerator and denominator by $P^\pi(y_{<t}^{(i)} \mid x, R_i = r_0)$ gives Eq. (24). If $k_c = 0$, the count constraint excludes outcome r_c, and both sides are zero. □

D.2 Critic Parameterization and Training

We represent the evidence with one shared critic. Let $h_\psi(s) \in \mathbb{R}^d$ be the representation produced by its LLM backbone. A single linear output head predicts the $C - 1$ nonreference coordinates:

$$\mathbf{f}_\phi(s) = \begin{bmatrix} 0 \\ W h_\psi(s) + b \end{bmatrix} \in \mathbb{R}^C, \qquad \phi = (\psi, W, b), \tag{25}$$

where $W \in \mathbb{R}^{(C-1)\times d}$ and $b \in \mathbb{R}^{C-1}$. The notation $f_{\phi,c}(s)$ refers to the c-th coordinate of $\mathbf{f}_\phi(s)$; all coordinates share the backbone and are computed in a single forward pass. Fixing $f_{\phi,0} \equiv 0$ removes the common additive offset, which leaves the normalized probabilities unchanged.

Write $R_i = r_{c_i}$. Substituting $f_{\phi,c}$ for Δ_c^π in Eq. (24) gives the negative log-likelihood objective

$$\mathcal{L}_{\text{critic}}^{(C)}(\phi) = -\mathbb{E}\left[\log \frac{K_{c_i} \exp\big(f_{\phi,c_i}(s_{i,t})\big)}{\sum_{\ell=0}^{C-1} K_\ell \exp\big(f_{\phi,\ell}(s_{i,t})\big)}\right]. \tag{26}$$

The expectation follows the sampling convention in Eq. (9). This objective jointly trains the shared backbone and output head using categorical cross-entropy, with logits $f_{\phi,c}(s_{i,t}) + \log K_c$ for classes with $K_c > 0$. Classes absent from the group are masked, and groups containing only one reward class contribute zero loss. As in the binary formulation, the group counts supply the class offsets, so the critic need not predict the prompt-level reward probabilities.

D.3 Value Reconstruction

For policy optimization, we combine the learned evidence with a smoothed leave-one-out estimate of the prompt-level reward distribution:

$$\widetilde{p}_{-i,c}(x) = \frac{\sum_{j \neq i} \mathbf{1}\{R_j = r_c\} + \alpha}{G - 1 + C\alpha}, \qquad \alpha > 0. \tag{27}$$

The pseudocount assigns positive prior probability to every reward class, including those absent from the sibling rollouts. Replacing p_c^π and Δ_c^π in Eq. (23) with $\widetilde{p}_{-i,c}$ and $f_{\phi,c}$, respectively, and taking the expected reward gives

$$V_\phi(s_{i,t}) = \frac{\sum_{c=0}^{C-1} r_c\, \widetilde{p}_{-i,c}(x) \exp\big(f_{\phi,c}(s_{i,t})\big)}{\sum_{c=0}^{C-1} \widetilde{p}_{-i,c}(x) \exp\big(f_{\phi,c}(s_{i,t})\big)}. \tag{28}$$

Under the terminal-reward setting with $\gamma = 1$, this provides the scalar value estimate required by GAE. The frozen estimates $\widehat{V}_{i,t} = V_{\phi_{\text{old}}}(s_{i,t})$ can therefore be used in the same PPO update as in the main text. Smoothing is applied only to the reconstruction prior; the training objective uses the observed counts K_c.

For $C = 2$ and $(r_0, r_1) = (0, 1)$, the head has one learned output, $K_1 = K$, and $K_0 = G - K$. Identifying $f_{\phi,1}$ with f_ϕ recovers the binary objective, prior estimator, and value reconstruction in Eqs. (9)–(11). We leave the corresponding reconstruction error analysis and empirical evaluation to future work.