UNCERTAINTY-GATED POLICY OPTIMIZATION FOR ROBUST REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent group-based Reinforcement Learning (RL) methods have greatly improved LLMs' reasoning capabilities. However, relying solely on sparse outcome-level rewards risks misaligned credit assignment: models may be rewarded for "lucky" trajectories that arrive at correct answers despite uncertain and potentially flawed intermediate reasoning. We introduce **U**ncertainty-**G**ated **A**dvantage (UGA), a plug-and-play mechanism that reshapes trajectory-level advantages using entropy-based uncertainty estimation. UGA directly dampens the advantage of high-uncertainty correct trajectories, reducing their contribution to policy updates while preserving learning signals from confident generations. The method requires no auxiliary reward model and integrates seamlessly into existing RL pipelines such as GRPO and DAPO. Experiments on challenging math benchmarks (AMC23, AIME24, AIME25) show that UGA consistently outperforms strong baselines across two model scales, achieving up to 11.6% relative improvement in mean@32 and 20.6% in maj@32. On knowledge-intensive agent tasks, UGA further improves accuracy by 5.0% (relative) over the baseline using step-level uncertainty gating.

1 INTRODUCTION

Recent advances in Large Language Models (LLMs) have shifted the focus toward "System 2" reasoning tasks, including mathematical problem solving (Uesato et al., 2022) and code generation. Although dense process rewards theoretically offer superior credit assignment by verifying intermediate steps (Lightman et al., 2023; Cui et al., 2025), successful applications in RL remain limited due to the challenges of acquiring scalable, high-quality signals. As a result, verifiable outcome-based rewards remain the cornerstone of state-of-the-art RL paradigms. Algorithms like Group Relative Policy Optimization (GRPO) (Shao et al., 2024) and DAPO (Yu et al., 2025) have achieved significant success by encouraging models to acquire reasoning behaviors through diverse trajectory sampling and outcome-driven optimization, despite the sparsity of the feedback signal.

However, a critical bottleneck of outcome-driven RL is that it treats the reasoning process as a black box, ignoring the reliability of the generation trajectory. Consequently, a correct output does not guarantee a sound reasoning chain, leading to unstable learning signals. In mathematical reasoning, models may arrive at the correct solution via erroneous steps or random guessing, creating false positive training samples. We provide supporting evidence for this association in Section 2.3: among verifier-accepted base-model solutions, process errors are substantially more common in high-entropy trajectories. Meanwhile, in knowledge-intensive agentic scenarios, models frequently produce plausible but fabricated intermediate steps characterized by high intrinsic uncertainty. By blindly reinforcing these trajectories based solely on the final outcome, current methods risk amplifying stochasticity and error, rather than cultivating robust reasoning capabilities.

We propose **Uncertainty-Gated Advantage (UGA)**, which reshapes outcome-based advantages using trajectory entropy and empirical problem difficulty. This requires addressing two challenges. First, problem difficulty changes as the policy learns and must therefore be estimated from the current policy's rollouts. Second, a uniform reshaping rule cannot account for the different implications of uncertainty across difficulty levels: high entropy may signal lucky guesses on easy problems but useful exploration on hard ones, so penalizing both equally can discard valuable learning signals.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To address the first challenge, we estimate difficulty online from the current policy's pass rate over a group of rollouts, with lower pass rates indicating harder problems. To address the second, we condition the entropy gate on this estimate: at the same entropy level, uncertain successes on easy problems receive less credit, while rare successes on hard problems retain more credit. This soft weighting preserves learning signals from confident generations while allowing uncertainty in successful attempts at difficult problems. The resulting gate requires no auxiliary reward model and integrates into GRPO and DAPO by modulating the advantage term.

We evaluate UGA on mathematical reasoning benchmarks (AMC23, AIME24, AIME25) and a knowledge-intensive Self-Search Agent task, demonstrating consistent improvements over strong RL baselines in both settings. Our main contributions are:

- We identify the "lucky guess" problem in outcome-driven RL for reasoning: high-uncertainty trajectories that arrive at correct answers receive undeserved credit, leading to noisy policy updates. A blinded process audit of verifier-accepted base-model solutions supports this view as an *enrichment* result: process errors are substantially more common in high-entropy traces.
- We propose UGA, a plug-and-play mechanism that reshapes advantages by combining entropy-based uncertainty estimation with difficulty-aware gating, requiring no auxiliary reward model and integrating directly into existing RL pipelines.
- Extensive experiments show that UGA consistently outperforms strong baselines across two model scales on math benchmarks, with gains transferring to cross-domain evaluations. On knowledge-intensive tasks, UGA further improves accuracy using step-level uncertainty gating.

2 Uncertainty-Gated Policy Optimization

2.1 Preliminary: RL for LLM Reasoning

Reinforcement learning has emerged as a powerful paradigm for enhancing the reasoning capabilities of LLMs. Given a question q, the model generates a complete response trajectory $\tau = (y_1, y_2, \ldots, y_T)$ autoregressively, and receives a scalar reward $r(\tau)$ based on the correctness of the final answer. The objective is to maximize the expected reward:

$$\mathcal{J}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\tau\sim\pi_\theta(\cdot|q)}\left[r(\tau)\right]. \tag{1}$$

GRPO. Group Relative Policy Optimization (Shao et al., 2024) samples a group of G responses $\{\tau_1, \ldots, \tau_G\}$ for each question q and computes advantages by normalizing rewards within the group:

$$A_i = \frac{r(\tau_i) - \mu_G}{\sigma_G}, \tag{2}$$

where μ_G and σ_G are the mean and standard deviation of rewards in the group. The policy is then updated using a clipped surrogate objective:

$$\mathcal{L}_{\text{GRPO}} = \mathbb{E}\left[\min\left(\rho A, \text{clip}(\rho, 1-\epsilon, 1+\epsilon)A\right)\right], \tag{3}$$

where $\rho = \frac{\pi_\theta(\tau|q)}{\pi_{\theta_{\text{old}}}(\tau|q)}$ is the importance sampling ratio.

DAPO. DAPO (Yu et al., 2025) introduces several refinements to GRPO, including Clip-Higher that uses asymmetric clipping bounds to encourage exploration, Dynamic Sampling that filters out questions where all responses are either correct or incorrect, and token-level loss computation. These modifications improve training stability.

While both methods have demonstrated strong results, they share a fundamental limitation: the advantage is computed solely based on outcome rewards, without considering the *reliability* of the generation process. This can lead to noisy credit assignment when correct answers arise from uncertain or spurious reasoning.

2.2 Uncertainty Measurement

To quantify the reliability of a generated trajectory, we measure the model's predictive uncertainty during generation. We adopt an entropy-based measure computed over the output distribution at each token position.

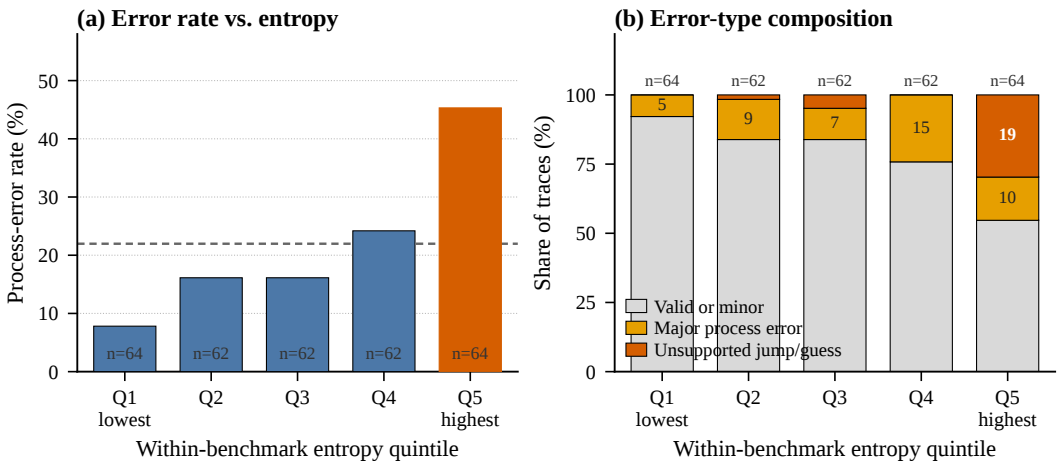

Figure 1: Process errors among verifier-accepted Qwen3-1.7B-Base solutions. Both panels use the same within-benchmark entropy quintiles. (a) Invalid-reasoning rate. The dashed line is the overall rate (22.0%). (b) Label composition. Counts inside segments are the number of traces. Invalid = major process error or unsupported jump/guess.

Specifically, for a trajectory τ generated in response to question q, we compute the token-level entropy at each position i using only the top-k tokens from the vocabulary:

$$H_i = -\sum_{j=1}^{k} p_\theta(y_i^{(j)}|q, y_{<i}) \log p_\theta(y_i^{(j)}|q, y_{<i}), \tag{4}$$

where $y_i^{(j)}$ denotes the j-th most probable token at position i. Using top-k tokens filters out noise from low-probability tokens.

The trajectory-level uncertainty $\mathcal{H}(\tau)$ is then computed as the mean entropy across all token positions:

$$\mathcal{H}(\tau) = \frac{1}{|\tau|} \sum_{i=1}^{|\tau|} H_i. \tag{5}$$

High values of $\mathcal{H}(\tau)$ indicate that the model exhibits substantial uncertainty throughout the generation, suggesting that the trajectory may rely on ambiguous reasoning or fortunate guessing rather than confident, well-grounded inference.

2.3 High-Entropy Correct Traces Are Enriched for Process Errors

The definition of $\mathcal{H}(\tau)$ is useful for advantage gating only if high entropy actually co-occurs with unreliable reasoning among trajectories that outcome rewards would treat as successes. We test this association on already generated evaluation rollouts from Qwen3-1.7B-Base on AMC23, AIME24, and AIME25—the same suites used in Section 4.1—restricted to solutions that an automatic verifier marks correct. Trajectory entropy uses the same top-k token entropy as in Eq. 5. A blinded LLM judge (Grok 4.6) then labels whether the reasoning substantively supports the answer, without access to entropy, model identity, or the purpose of the study. We calibrate this critic on the full Process-Bench dump (Zheng et al., 2025): among solutions whose final answer is already correct, it detects process errors with binary F1 0.862 (Appendix F). We treat *major process errors* and *unsupported jumps or guesses* as process-invalid; uncertain labels are excluded. Entropy quintiles are assigned within each benchmark *before* judging. The primary analysis contains 314 traces from 41 questions (290 of them from AMC23). This diagnostic is correlational: it does not claim that entropy detects errors, nor that training with UGA reduces them.

Figure 1(a) shows that process-invalid rates rise with within-benchmark entropy percentile and jump in the upper tail: 7.8% in the lowest quintile versus 45.3% in the highest. Figure 1(b) further shows

that unsupported jumps and guesses—the closest analogue of lucky guesses—are almost confined to the highest quintile (19 of 23 such traces). Major process errors occur in every quintile. Low-entropy traces still contain process errors, so $\mathcal{H}(\tau)$ is a noisy but informative proxy that *enriches* for process-unreliable correct trajectories rather than identifying them.

These observations motivate a soft, continuous gate rather than a hard filter. Outcome-only advantages would otherwise assign full positive credit to a subset of high-uncertainty, process-unreliable successes, while a binary cutoff would also discard some valid high-entropy solutions and would ignore residual errors among confident traces. Two annotated traces appear in Appendix F.

2.4 Uncertainty-Gated Advantage

Given this association, we propose a gating mechanism that modulates the advantage according to the model's confidence. High-entropy trajectories that arrive at correct answers are more often process-unreliable, and should receive reduced—but not necessarily zero—credit.

Gating Function. We define a general gating function that maps trajectory-level uncertainty to a scaling factor:

$$\psi(\tau) = \frac{1}{1 + \beta \cdot f(\mathcal{H}(\tau))}, \tag{6}$$

where $\beta > 0$ is a scaling hyperparameter and $f(\cdot)$ is a mapping function that transforms the raw entropy into a gating signal. This formulation ensures that ψ decreases monotonically as H increases. For the mapping function $f(\cdot)$, we consider two variants: (1) **identity mapping** $f(x) = x$ for linear sensitivity, and (2) **sigmoid mapping** $f(x) = \sigma(\kappa(x - \mu))$ for non-linear soft thresholding.

Difficulty-Aware Extension. A key insight is that uncertainty should be interpreted *relative to problem difficulty*. High entropy on genuinely difficult problems is expected and should not be heavily penalized, while high entropy on easy problems signals potential lucky guesses.

To incorporate difficulty awareness, we extend the gating function by introducing the pass rate $p(q)$ as a difficulty proxy. For each question q, the pass rate is computed as:

$$p(q) = \frac{1}{G} \sum_{i=1}^{G} \mathbf{1}[r(\tau_i) = 1], \tag{7}$$

where G is the group size. A high pass rate indicates an easy problem for the current policy, while a low pass rate suggests a difficult one. The difficulty-aware gating function becomes:

$$\psi(\tau, q) = \frac{1}{1 + \beta \cdot \dfrac{f(\mathcal{H}(\tau))}{(r(\tau) - p(q))^2}}, \tag{8}$$

where $r(\tau) \in \{0, 1\}$ is the binary reward. The denominator $(r(\tau) - p(q))^2$ modulates the gate based on outcome and difficulty:

- **Correct Trajectory** ($r = 1$): denominator is $(1 - p(q))^2$, which is small on easy problems, suppressing high-entropy lucky guesses, and large on hard problems, preserving credit under moderate uncertainty.
- **Incorrect Trajectory** ($r = 0$): denominator is $p(q)^2$, which is large on easy problems, retaining more of the negative penalty where failure is less excusable.

Reshaped Advantage. The final uncertainty-gated advantage is obtained by scaling the original advantage:

$$\tilde{A}(\tau) = \psi(\tau, q) \cdot A(\tau). \tag{9}$$

Importantly, the gating term $\psi(\tau, q)$ is detached from the computational graph, acting purely as a fixed scaling factor during backpropagation.

Integration with Existing Methods. UGA can be seamlessly integrated into GRPO or DAPO by simply replacing A with $\tilde{A}$ in the policy gradient computation. No additional reward model or architectural changes are required. The complete training procedure is given in Algorithm 1 (Appendix A).

Computational Cost. UGA uses the policy's token probabilities to estimate top-k entropy, adding approximately 10% to training time in our experiments. The detached gate rescales the base advantage without adding a learned reward model or an auxiliary training loss.

3 Extension to Self-Search Agent

In Self-Search Agents (Fan et al., 2025), the model interleaves reasoning actions with self-generated observations (predicted search results), without external retrieval. A trajectory therefore contains both reasoning and world-model steps. We adapt UGA to this finer granularity by measuring the mean top-k token entropy H_t of each observation, using the estimator in Section 2.2.

Knowledge-intensive generations can admit multiple valid realizations, so we use a sigmoid mapping rather than the identity mapping used for math. The observation gate and reshaped world-model advantage are

$$\begin{aligned} \psi(H_t) &= \left[1 + \beta\, \sigma(\kappa(H_t - \mu))\right]^{-1}, \\ \tilde{A}_t^w &= \begin{cases} [1 - w + w\psi(H_t)]A_t^w, & A_t^w > 0, \\ A_t^w, & A_t^w \le 0. \end{cases} \end{aligned} \tag{10}$$

Here μ sets a soft entropy threshold, κ controls its sharpness, and $w \in [0, 1]$ preserves a fraction of the original positive advantage. Unlike the math gate, this agent variant leaves negative advantages unchanged, retaining the full penalty for failures.

Reasoning actions also depend on the reliability of preceding observations. We therefore propagate the preceding observation's gate to the positive reasoning advantage:

$$\tilde{A}_t^r = \begin{cases} \gamma\psi(H_{t-1})A_t^r, & t > 0 \text{ and } A_t^r > 0, \\ A_t^r, & \text{otherwise}, \end{cases} \tag{11}$$

where γ is a discount factor. Initial reasoning and nonpositive advantages remain unchanged. Both components enter the standard policy update with detached gates. Appendix B gives the full trajectory factorization, step-level objectives, and combined gradient.

4 Experiments

We evaluate UGA in two settings: (1) mathematical reasoning, where we test whether uncertainty-gated optimization improves accuracy on challenging benchmarks, and (2) a Self-Search Agent task, where we measure final-answer accuracy in knowledge-intensive scenarios. All experiments use the verl[1] framework for RL training.

4.1 Mathematical Reasoning

Setup. We evaluate on three mathematical reasoning benchmarks: AMC23, AIME24 and AIME25. We apply UGA on top of two strong RL baselines: GRPO (Shao et al., 2024) and DAPO (Yu et al., 2025), using Qwen3-1.7B-Base and Qwen3-4B-Base as base models. We further compare against recent RL baselines, including ERPO (Yu et al., 2026), GRPO-LEAD (Zhang & Zuo, 2025), and S-GRPO (Shen et al., 2025).

For training, we use the DAPO-Math-17k dataset and train for approximately 100 steps. The training hyperparameters are set as: $\beta = 0.3$, $k = 50$, with learning rate 10^{-6} and batch size 512. For evaluation, we sample 32 responses per question with temperature = 0.7 and top-p = 0.8. Here mean@32 averages correctness over the 32 sampled responses, whereas maj@32 scores the answer selected by majority voting.

[1] https://github.com/volcengine/verl

Model	AMC23		AIME24		AIME25		Average	
	mean	maj	mean	maj	mean	maj	mean	maj
Qwen3-1.7B-Base	23.59	27.50	2.92	3.33	1.88	0.00	9.46	10.28
+GRPO	43.36	47.50	12.29	23.33	8.02	10.00	21.22	26.94
+GRPO w/ UGA	47.97	57.50	13.75	20.00	9.38	20.00	23.70	32.50
Δ	+4.61	+10.00	+1.46	-3.33	+1.35	+10.00	+2.47	+5.56
+DAPO	45.47	55.00	12.19	16.67	8.33	10.00	22.00	27.22
+DAPO w/ UGA	47.03	52.50	14.69	20.00	9.69	13.33	23.80	28.61
Δ	+1.56	-2.50	+2.50	+3.33	+1.35	+3.33	+1.81	+1.39
Qwen3-4B-Base	36.02	40.00	7.60	13.33	8.44	13.33	17.35	22.22
+GRPO	63.36	70.00	23.65	33.33	20.21	26.67	35.74	43.33
+GRPO w/ UGA	62.66	77.50	26.25	36.67	22.81	33.33	37.24	49.17
Δ	-0.70	+7.50	+2.60	+3.33	+2.60	+6.67	+1.50	+5.83
+DAPO	67.11	80.00	29.38	36.67	25.52	30.00	40.67	48.89
+DAPO w/ UGA	69.92	82.50	31.15	40.00	26.98	33.33	42.68	51.94
Δ	+2.81	+2.50	+1.77	+3.33	+1.46	+3.33	+2.01	+3.06

Table 1: Main results on mathematical reasoning benchmarks. We report mean@32 and maj@32 accuracy (%). Δ denotes performance gaps compared to the baseline without UGA. Base-model AIME24/25 scores use Math-Verify, which is more tolerant of unboxed answers than the string matcher used for the RL checkpoints.

Method	AMC23		AIME24		AIME25		Average	
	mean	maj	mean	maj	mean	maj	mean	maj
GRPO	43.36	47.50	12.29	**23.33**	8.02	10.00	21.22	26.94
ERPO	47.11	**57.50**	11.77	16.67	7.50	16.67	22.13	30.28
GRPO-LEAD	**48.83**	**57.50**	**13.85**	16.67	8.02	13.33	23.57	29.17
S-GRPO	44.38	55.00	13.44	20.00	9.17	**20.00**	22.33	31.67
GRPO w/ UGA	47.97	**57.50**	13.75	20.00	**9.38**	**20.00**	**23.70**	**32.50**

Table 2: Comparison with recent RL baselines on Qwen3-1.7B-Base under the same GRPO training settings, data, and evaluation protocol. Columns report mean@32 and maj@32 accuracy (%).

Results. Table 1 presents the main results. UGA consistently improves performance across all experimental configurations, achieving mean@32 gains of +1.50 to +2.47 and maj@32 gains up to +5.83 over baselines. The improvements under maj@32 are particularly notable, as majority voting amplifies the benefit of more consistent and confident reasoning, consistent with more reliable aggregation across sampled answers.

An interesting observation is that UGA yields larger majority voting gains when applied to GRPO compared to DAPO across both model scales. This is likely because DAPO already incorporates Clip-Higher, an asymmetric clipping mechanism that also operates the advantage, resulting in a stronger baseline that may leave less room for further optimization. Nevertheless, UGA still provides consistent gains on top of DAPO, demonstrating that uncertainty-aware reweighting offers complementary benefits.

Table 2 further compares UGA with recent RL baselines under the same Qwen3-1.7B-Base GRPO setting. ERPO uses entropy-aware token-level gating to emphasize critical decision pivots, GRPO-LEAD reweights advantages according to empirical problem difficulty, and S-GRPO applies noise-aware advantage reweighting to mitigate think-answer mismatch in group-relative rewards. In contrast to these single-axis reweighting strategies, UGA treats uncertainty as a reliability signal whose effect depends on both problem-level pass rate and the direction of the advantage. The results show that this gating achieves the best average mean@32 and maj@32. The mean@32 margin over GRPO-LEAD is small (23.70 versus 23.57), and improvements are not uniform across benchmarks.

We further examine pass@k performance in Appendix E. Gains are generally stronger at small k, while some configurations decline at larger k. Thus, improved average accuracy does not imply uniformly improved coverage of correct solutions.

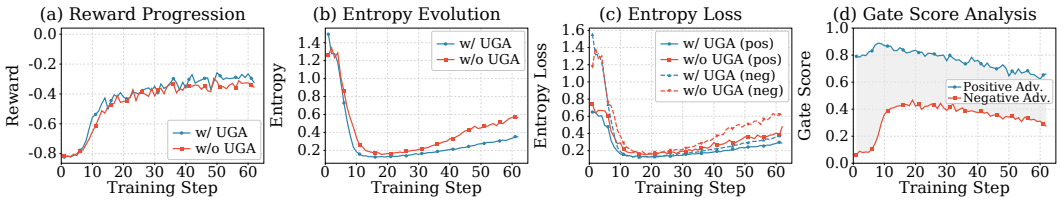

Figure 2: Training dynamics comparison between w/ UGA and w/o UGA. (a) Reward progression (b) Entropy evolution (c) Entropy loss decomposition by advantage sign (d) Gate score analysis

Method	General QA		Multi-Hop QA				Avg
	NQ	TQ	HotpotQA	Musique	2wiki	Bamboogle	
ZeroSearch-Instruct	40.2	58.0	22.8	10.4	21.4	18.5	28.5
ZeroSearch-Base	43.4	**63.8**	**32.2**	13.8	**35.6**	15.3	34.0
SSRL w/ info mask	43.8	58.4	25.0	**14.2**	31.6	38.4	35.2
SSRL w/o info mask	43.4	57.2	25.0	11.8	28.8	38.4	34.1
Ours	**46.0**	59.2	26.0	13.6	30.0	**40.0**	**35.8**

Table 3: Performance comparison on General QA and Multi-Hop QA datasets.

Cross-Domain Transfer. We additionally evaluate all math-trained checkpoints on MMLU and MMLU-Pro to test zero-shot cross-domain transfer. Results in Appendix C show that UGA improves performance in 3 out of 4 configurations, indicating benefits in most tested configurations, with a small decrease for GRPO 1.7B.

4.2 Training Dynamics

To understand how UGA affects the learning process, we analyze the training dynamics through three perspectives: reward progression, entropy evolution, and gate score distribution. Figure 2 summarizes these analyses.

Reward Progression. As shown in Figure 2(a), UGA achieves relatively higher rewards compared to the baseline throughout training. The gap emerges early and persists, demonstrating that by down-weighting noisy training samples, UGA enables more effective gradient updates that focus on reliable trajectories.

Entropy Evolution. Figure 2(b) reveals a characteristic two-phase entropy pattern. In the early phase, both methods exhibit rapid entropy decrease as the model focuses its sampling distribution. In the later phase, entropy gradually increases. Crucially, UGA moderates this entropy increase, preserving sufficient exploration while preventing excessive entropy that would lead to inconsistent outputs. The entropy loss decomposition in Figure 2(c) further shows that UGA reduces entropy loss magnitudes for both positive and negative advantage trajectories.

Gate Score Analysis. Figure 2(d) shows that trajectories with positive advantages consistently receive higher gate scores than those with negative advantages. This is consistent with asymmetric weighting of successful and unsuccessful trajectories, although the aggregate curves do not isolate the effects of problem difficulty.

4.3 Self-Search Agent

Setup. To evaluate UGA in agentic scenarios, we use the Self-Search Agent task following the setup of Fan et al. (2025). In this setting, the model must answer knowledge-intensive questions by interleaving reasoning with self-generated "search results" rather than using an external retrieval.

Following Fan et al. (2025), we use Llama-3.2-3B-Instruct as the base model and train on a combination of NQ (Kwiatkowski et al., 2019) and HotpotQA (Yang et al., 2018) training sets. We set the

group size $G = 5$, learning rate 10^{-6}, and batch size 256. The step-level gating hyperparameters are: $\mu = 0.6$, $\kappa = 1$, $w = 0.3$, and $\gamma = 0.5$.

We evaluate on six knowledge QA benchmarks: NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023), reporting answer accuracy.

Results. Table 3 presents the results. Following the terminology of Fan et al. (2025), "mask" here refers to the *information mask* that controls whether the world-model component participates in gradient updates. Our primary baseline is **SSRL w/o info mask**, which allows the world-model component (search result generation) to be optimized jointly. In contrast, **SSRL w/ info mask** completely excludes the world-model gradient, removing all observation-step updates. This differs from our soft gate, which retains both a positive-advantage floor and full negative advantages.

The results reveal an informative pattern. SSRL w/o info mask (34.1%) underperforms SSRL w/ info mask (35.2%), suggesting that naively optimizing the world-model component introduces noise from hallucinated observations. However, UGA (35.8%) outperforms both, demonstrating that **dynamic uncertainty-aware gating provides a better trade-off than binary inclusion or exclusion**. Rather than completely blocking world-model optimization, UGA selectively down-weights high-entropy steps while preserving learning signals from confident generations.

Configuration	Gating Factor	AMC23		AIME24		AIME25		Avg.		Δ	
		mean	maj	mean	maj	mean	maj	mean	maj	mean	maj
UGA (Ours)	$\frac{1}{1+\beta\frac{\mathcal{H}}{(r-p)^2}}$	**47.97**	57.50	**13.75**	20.00	**9.38**	**20.00**	**23.70**	32.50	-	-
Ablation on Uncertainty Metric											
w/o Instance Selectivity	$\mathcal{H} \leftarrow \bar{\mathcal{H}}_{\text{batch}}$	43.98	57.50	11.77	16.67	6.98	13.33	20.91	29.17	-2.79	-3.33
w/o Uncertainty	$\mathcal{H} \leftarrow 0.2$	45.47	52.50	11.15	16.67	6.46	10.00	21.02	26.39	-2.68	-6.11
Ablation on Difficulty Awareness											
w/o Pass Rate	$p(q) \leftarrow 0.5$	47.66	**65.00**	12.81	20.00	8.75	16.67	23.07	**33.89**	-0.63	+1.39
Inverted Pass Rate	$r(\tau) \leftarrow 1 - r(\tau)$	46.48	55.00	8.54	16.67	6.88	10.00	20.63	27.22	-3.07	-5.28
Baseline											
GRPO	Gate $\equiv 1$	43.36	47.50	12.29	**23.33**	8.02	10.00	21.22	26.94	-2.48	-5.56

Table 4: Ablation study on mathematical reasoning benchmarks using Qwen3-1.7B-Base with GRPO. We report mean@32 and maj@32 accuracy (%) on AMC23, AIME24, and AIME25. The Δ columns indicate the difference in average accuracy compared to the full UGA method.

4.4 Ablation Study

We conduct ablation studies to verify the contribution of each component in UGA. All ablations are performed on the mathematical reasoning task using Qwen3-1.7B-Base with GRPO as the base method. Results are shown in Table 4.

Impact of Uncertainty Metric. To verify that the effectiveness of UGA stems from distinguishing instance-level uncertainty rather than merely applying a global penalty, we replace the trajectory-specific entropy $\mathcal{H}(\tau)$ with the batch-level average entropy $\bar{\mathcal{H}}_{\text{batch}}$ ("w/o Instance Selectivity") or a fixed constant ("w/o Uncertainty"). Both ablations result in significant performance drops, demonstrating that per-trajectory uncertainty measurement is essential. The variance in uncertainty across different trajectories carries meaningful information that cannot be captured by a single batch-level statistic or constant value. Notably, the degradation is more pronounced under maj@32, suggesting that instance-level uncertainty signals are particularly important for maintaining consistent outputs across multiple samples.

Impact of Difficulty Awareness Two ablations test the difficulty-aware term $(r(\tau) - p(q))^2$. Setting a constant pass-rate $p(q) = 0.5$ ("w/o Pass Rate") results in a modest decrease in mean@32,

while inverting the pass-rate ("Inverted Pass Rate") leads to substantially larger degradation. This contrast highlights the importance of the weighting direction: the inverted gate preserves more credit for easy-problem successes while suppressing rare successes on hard problems.

The fixed-pass-rate variant achieves higher maj@32 (33.89 versus 32.50), but lower mean@32 (23.07 versus 23.70), revealing a trade-off between single-sample and voting accuracy.

A sensitivity analysis on the hyperparameter β is presented in Appendix D.

5 Related Work

Reinforcement Learning for LLM Alignment. Reinforcement learning has become a cornerstone for aligning LLMs with human preferences. RLHF (Ouyang et al., 2022) pioneered the use of PPO (Schulman et al., 2017) to optimize LLMs against learned reward models. Recent work has proposed more efficient alternatives: DPO (Rafailov et al., 2023) eliminates the need for explicit reward modeling by directly optimizing preferences, while GRPO (Shao et al., 2024) introduces group-relative advantage estimation that removes the critic network. DAPO (Yu et al., 2025) further improves training stability through dynamic sampling and asymmetric clipping. However, these methods compute advantages solely based on outcome rewards, without considering the reliability of the generation process. Our work addresses this limitation by incorporating uncertainty-aware advantage reweighting.

Uncertainty Quantification in LLMs. Estimating uncertainty in LLM outputs is crucial for reliable deployment. Entropy-based measures (Malinin & Gales, 2018; Xiao et al., 2022) quantify uncertainty through the spread of output distributions. Recent work has explored leveraging uncertainty for improved reasoning: Cheng et al. (2025) observe that high-entropy regions correlate with exploratory reasoning actions and augment advantages with an entropy bonus to encourage exploration. In contrast, Prabhudesai et al. (2025) use confidence (inverse entropy) directly as a reward signal. While both approaches recognize the value of uncertainty signals, they aim to *promote* high-entropy behaviors for exploration or reward confident outputs. Our method takes the opposite perspective: we *down-weight* high-entropy trajectories that yield correct answers, viewing such cases as potential lucky guesses rather than genuine reasoning.

Reward Shaping and Credit Assignment. Effective credit assignment remains challenging in RL for reasoning tasks. Process Reward Models (PRMs) (Lightman et al., 2023; Wang et al., 2024) address this by providing step-level supervision, but require expensive human annotations. Outcome Reward Models (ORMs) are simpler but suffer from noisy credit assignment when spurious trajectories receive positive rewards. Recent work has explored automated process supervision (Luo et al., 2024) and self-verification (Weng et al., 2023) to improve credit assignment without manual labels. Recent RL baselines improve GRPO-style credit assignment from complementary perspectives: ERPO emphasizes high-entropy decision points (Yu et al., 2026), GRPO-LEAD increases updates on difficult problems (Zhang & Zuo, 2025), and S-GRPO attenuates noisy reward groups (Shen et al., 2025). Our approach offers a complementary solution: rather than training additional reward models or relying on a single reshaping signal, we use the policy's own uncertainty together with problem difficulty and advantage direction as a proxy for trajectory reliability, enabling fine-grained advantage reweighting within standard RL pipelines.

6 Conclusion

We introduce UGA, an advantage-reweighting mechanism that combines trajectory entropy with empirical problem difficulty, and extend it to step-level agent optimization. A blinded audit finds that high-entropy, verifier-accepted base-model traces are enriched for process errors, motivating a soft reliability gate. UGA improves average mathematical reasoning accuracy across GRPO and DAPO at two model scales and improves knowledge QA accuracy, without an auxiliary reward model. The gate preserves the sign of the base advantage and adjusts its magnitude, offering a continuous alternative to accepting or discarding uncertain generations outright. These results support uncertainty-aware credit assignment, while measuring its causal effect on process reliability remains future work.

REFERENCES

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *CoRR*, abs/2502.01456, 2025.

Yuchen Fan, Kaiyan Zhang, Heng Zhou, Yuxin Zuo, Yanxu Chen, Yu Fu, Xinwei Long, Xuekai Zhu, Che Jiang, Yuchen Zhang, et al. Ssrl: Self-search reinforcement learning. *arXiv preprint arXiv:2508.10874*, 2025.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*, 2020.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, et al. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*, 2024.

Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems*, 31, 2018.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Maximizing confidence alone improves reasoning. *arXiv preprint arXiv:2505.22660*, 2025.

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, 2023.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

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, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Si Shen, Peijun Shen, Wenhua Zhao, and Danhao Zhu. Mitigating think-answer mismatch in llm reasoning through noise-aware advantage reweighting. *arXiv preprint arXiv:2508.05928*, 2025.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 2022.

Jonathan Uesato, Nate Kushman, Ramana Kumar, H. Francis Song, Noah Y. Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *CoRR*, abs/2211.14275, 2022.

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, 2024.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2550–2575, 2023.

Yuxin Xiao, Paul Pu Liang, Umang Bhatt, Willie Neiswanger, Ruslan Salakhutdinov, and Louis-Philippe Morency. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. *arXiv preprint arXiv:2210.04714*, 2022.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William 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, 2018.

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.

Song Yu, Li Li, Wenwen Zhao, and Zhisheng Yang. Erpo: Token-level entropy-regulated policy optimization for large reasoning models. *arXiv preprint arXiv:2603.28204*, 2026.

Jixiao Zhang and Chunsheng Zuo. Grpo-lead: A difficulty-aware reinforcement learning approach for concise mathematical reasoning in language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5642–5665, 2025.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. ProcessBench: Identifying process errors in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1009–1024, 2025.

A Training Algorithm

Algorithm 1 summarizes trajectory-level UGA. The gate is detached before the policy update; the underlying GRPO or DAPO loss is otherwise retained.

Algorithm 1 UGA-Enhanced Policy Optimization

Require: Dataset $\mathcal{D}$, policy π_θ, group size G, hyperparameters β, k

for each training iteration **do**
  Sample batch of questions $\{q\}$ from $\mathcal{D}$
  for each question q **do**
    Sample G trajectories $\{\tau_1, \ldots, \tau_G\}$ from $\pi_\theta(\cdot|q)$
    Compute rewards $\{r(\tau_i)\}$ and pass rate $p(q)$
    Compute base advantages $\{A(\tau_i)\}$ via GRPO/DAPO
    for each trajectory τ_i **do**
      Compute entropy $\mathcal{H}(\tau_i)$ using Eq. 5
      Compute gate $\psi(\tau_i, q)$ using Eq. 8
      Compute $\tilde{A}(\tau_i) = \psi(\tau_i, q) \cdot A(\tau_i)$
    end for
  end for
  Update θ using policy gradient with $\tilde{A}$
end for

B Self-Search Agent: Full Formulation

We now extend UGA to a **Self-Search Agent** setting, where the model interleaves reasoning with knowledge generation. This formulation extends uncertainty-gated optimization beyond single-turn reasoning; the experiments evaluate final-answer accuracy.

B.1 Problem Formulation

We formalize the agent's decision-making as a Markov Decision Process. At each time step t, the agent in state $s_t \in \mathcal{S}$ chooses an action $a_t \in \mathcal{A}$ according to its policy $\pi(a_t \mid s_t)$. The environment then emits an observation $o_{t+1} \in \mathcal{O}$ and transitions to state s_{t+1}. The trajectory can be expressed as $\tau = (s_0, a_0, o_1, s_1, a_1, o_2, \ldots, s_T)$, with probability:

$$P(\tau) = \prod_{t=0}^{T-1} \underbrace{P_r(a_t \mid s_t)}_{\text{Reasoning}} \cdot \underbrace{P_w(o_{t+1} \mid s_t, a_t)}_{\text{World Model}}, \tag{12}$$

where P_r is the reasoning policy and P_w generates observations (e.g., predicted search results). The RL objective follows a PPO-style formulation with step-level advantages:

$$\max_{\pi_\theta} \mathbb{E}\left\{\sum_{t=0}^{T-1} \min\left[\rho_t(\theta)A_t, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right]\right\}, \tag{13}$$

where A_t can be decomposed into A_t^r (for reasoning actions) and A_t^w (for world-model observations).

B.2 Adapting UGA to Step-Level Optimization

Applying the trajectory-level UGA (Eq. 6) directly to the agent setting faces two key challenges:

Challenge 1: Granularity Mismatch. In mathematical reasoning, the entire trajectory receives a single reward, making trajectory-level gating natural. In contrast, agent trajectories contain multiple action-observation steps, each potentially contributing differently to the final outcome.

Challenge 2: Uncertainty Calibration. Mathematical reasoning typically follows relatively structured solution paths, where entropy can be directly used as the gating signal (i.e., identity mapping). However, knowledge-intensive tasks exhibit greater variability: the same query may admit multiple valid responses (e.g., range queries or open-ended factual questions), leading to a more dispersed entropy distribution. This necessitates a softer mapping function that establishes a threshold to distinguish genuinely uncertain (potentially hallucinated) generations from acceptable variance.

Step-Level Gating. To address these challenges, we adapt UGA to operate at the step level. For each observation o_t, we compute the step-level entropy as the mean token entropy:

$$H_t = -\frac{1}{|o_t|} \sum_{i=1}^{|o_t|} \sum_{j=1}^{k} p_\theta^{(j)} \log p_\theta^{(j)}, \tag{14}$$

where $p_\theta^{(j)} = p_\theta(o_{t,i}^{(j)} | s_t, a_t, o_{t,<i})$ denotes the probability of the j-th most likely token at position i.

Following the base gating function (Eq. 6), we define the step-level gate as:

$$\psi(\tau) = \frac{1}{1 + \beta \cdot \sigma(\kappa(H_t - \mu))}, \tag{15}$$

where we select a sigmoid mapping $f(H_t) = \sigma(\kappa(H_t - \mu))$ that applies a soft threshold at μ with sharpness κ. This will suppress ψ as entropy exceeds an acceptable threshold μ.

Asymmetric Advantage Reshaping. A critical observation is that gating should be *asymmetric* with respect to the sign of the advantage. For positive advantages, we want to down-weight high-uncertainty steps to avoid reinforcing lucky guesses. However, for negative advantages, applying the same gate would reduce the penalty magnitude, effectively weakening the learning signal from failures. Therefore, we apply gating only to positive advantages:

$$\tilde{A}_t^w = \begin{cases} w \cdot \psi(H_t) \cdot A_t^w + (1 - w) \cdot A_t^w, & \text{if } A_t^w > 0 \\ A_t^w, & \text{otherwise} \end{cases} \tag{16}$$

where $w \in [0, 1]$ controls the strength of entropy-based adjustment. The term $(1 - w)$ ensures a portion of the original advantage is always preserved, providing a soft rather than hard gating effect.

B.3 Extension to Reasoning Steps

The interleaved nature of agent trajectories creates dependencies: reasoning decisions at step t rely on the observation o_t from the previous world-model step. If o_t is hallucinated, subsequent reasoning may be misguided.

To mitigate this error propagation, we extend the gating mechanism to reasoning steps. The reshaped advantage for reasoning at step t inherits the uncertainty from the preceding observation:

$$\tilde{A}_t^r = \begin{cases} A_t^r, & \text{if } t = 0 \\ \gamma \cdot \psi(H_{t-1}) \cdot A_t^r, & \text{if } t > 0 \text{ and } A_t^r > 0 \\ A_t^r, & \text{otherwise} \end{cases} \tag{17}$$

where γ is a discount factor and H_{t-1} is the entropy of the preceding observation. This ensures that reasoning steps conditioned on uncertain observations receive appropriately reduced credit.

Unified Gradient Update. The final gradient combines both components:

$$\begin{aligned} \nabla_\theta \mathcal{J} = \mathbb{E}\Big[& \sum_{t=0}^{T-1} \tilde{A}_t^r \nabla_\theta \log \pi_\theta(a_t | s_t) \\ & + \sum_{t=0}^{T-1} \tilde{A}_t^w \nabla_\theta \log \pi_\theta(o_{t+1} | s_t, a_t) \Big]. \end{aligned} \tag{18}$$

C Cross-Domain Transfer Results

To assess whether UGA's improvements on mathematical reasoning generalize to other domains, we evaluate all math-trained checkpoints from Section 4.1 on MMLU and MMLU-Pro, which together cover 50+ subjects including physics, chemistry, engineering, health, biology, and history. Crucially, no domain-specific fine-tuning is performed; we directly evaluate the checkpoints trained on math data, so these results reflect zero-shot cross-domain transfer ability.

As shown in Table 5, 3 out of 4 configurations (DAPO 1.7B, GRPO 4B, DAPO 4B) show consistent improvements with UGA. These results show improved cross-domain transfer in most tested configurations, although GRPO 1.7B exhibits a small decrease.

Configuration	MMLU		MMLU-Pro	
	mean@8	maj@8	mean@8	maj@8
Qwen3-1.7B-Base	38.70	46.47	20.72	24.96
+GRPO	57.61	62.03	39.53	44.88
+GRPO w/ UGA	57.26	60.51	38.51	43.45
+DAPO	55.39	61.13	37.18	43.38
+DAPO w/ UGA	**56.87**	**62.93**	**38.35**	**44.70**
Qwen3-4B-Base	66.24	70.77	45.22	53.06
+GRPO	72.65	74.78	57.86	61.86
+GRPO w/ UGA	**73.55**	**75.72**	**59.27**	**63.29**
+DAPO	72.83	75.85	57.97	62.89
+DAPO w/ UGA	**73.20**	**76.16**	**59.05**	**64.35**

Table 5: Zero-shot cross-domain transfer results. All models are trained exclusively on math data and evaluated on MMLU and MMLU-Pro without any domain-specific fine-tuning. We report mean@8 and maj@8 accuracy (%).

D Hyperparameter Sensitivity

UGA introduces one key hyperparameter β beyond the standard GRPO/DAPO setup. We conduct sensitivity experiments on β using the GRPO 1.7B configuration. As shown in Table 6, all tested β values outperform the GRPO baseline on both mean@32 and maj@32. The default $\beta = 0.3$ achieves the best overall performance, while $\beta = 0.4$ shows diminished gains, suggesting that overly aggressive suppression starts to hurt. Within this range, UGA performs stably and consistently above baseline.

β	AMC23		AIME24		AIME25		Average	
	mean@32	maj@32	mean@32	maj@32	mean@32	maj@32	mean@32	maj@32
GRPO baseline	43.36	47.50	12.29	23.33	8.02	10.00	21.22	26.94
0.2	44.45	57.50	11.98	20.00	8.85	13.33	21.76	30.28
0.3 (default)	**47.97**	**57.50**	**13.75**	20.00	**9.38**	**20.00**	**23.70**	**32.50**
0.4	45.16	55.00	12.60	20.00	5.94	10.00	21.23	28.33

Table 6: Sensitivity analysis on β using Qwen3-1.7B-Base with GRPO. Note that $\beta = 0$ corresponds to the GRPO baseline (no gating, $\psi \equiv 1$), and increasing β strengthens the uncertainty penalty. All tested β values outperform the baseline on both mean@32 and maj@32. The default $\beta = 0.3$ achieves the best overall performance.

E Detailed Pass@k Results

We provide detailed pass@k results for mathematical reasoning benchmarks in Table 7, Table 8, and Table 9. These tables report accuracy across different values of k (1, 2, 4, 8, 16, 32), where pass@k measures the probability that at least one of k sampled solutions is correct. As in Table 1, base-model AIME24/25 scores use Math-Verify. For each problem, we sample $n = 32$ solutions

and compute pass@k using the unbiased estimator (Chen et al., 2021):

$$\text{pass@}k = 1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}, \tag{19}$$

where c is the number of correct solutions among n samples.

Model	pass@1	pass@2	pass@4	pass@8	pass@16	pass@32
Qwen3-1.7B-Base	23.59	36.36	49.63	60.92	69.31	77.50
+GRPO	43.36	54.69	64.18	72.90	81.09	87.50
+GRPO w/ UGA	47.97	57.92	66.36	75.22	83.23	87.50
Δ	+4.61	+3.24	+2.17	+2.31	+2.14	0.00
+DAPO	45.47	56.21	65.49	74.11	80.94	85.00
+DAPO w/ UGA	47.03	56.79	65.25	73.97	82.37	87.50
Δ	+1.56	+0.58	-0.24	-0.14	+1.43	+2.50
Qwen3-4B-Base	36.02	51.10	65.69	78.00	85.43	90.00
+GRPO	63.36	74.91	83.65	89.99	93.36	95.00
+GRPO w/ UGA	62.66	73.80	81.99	87.15	90.30	92.50
Δ	-0.70	-1.11	-1.66	-2.84	-3.06	-2.50
+DAPO	67.11	79.38	87.42	92.29	95.54	97.50
+DAPO w/ UGA	69.92	81.20	87.56	91.25	93.49	95.00
Δ	+2.81	+1.82	+0.13	-1.04	-2.04	-2.50

Table 7: Pass@k accuracy (%) on AMC23. Δ denotes performance gaps compared to the baseline without UGA.

Model	pass@1	pass@2	pass@4	pass@8	pass@16	pass@32
Qwen3-1.7B-Base	2.92	5.14	8.42	12.95	19.08	26.67
+GRPO	12.29	17.53	21.92	24.99	27.50	30.00
+GRPO w/ UGA	13.75	18.41	21.96	24.27	26.59	30.00
Δ	+1.46	+0.88	+0.04	-0.72	-0.90	0.00
+DAPO	12.19	17.00	21.10	24.45	28.39	33.33
+DAPO w/ UGA	14.69	19.40	23.58	27.24	30.62	33.33
Δ	+2.50	+2.40	+2.48	+2.78	+2.23	0.00
Qwen3-4B-Base	7.60	12.02	17.08	22.57	28.85	36.67
+GRPO	23.65	30.28	36.71	42.45	47.98	53.33
+GRPO w/ UGA	26.25	33.84	40.97	47.58	53.84	60.00
Δ	+2.60	+3.56	+4.26	+5.13	+5.86	+6.67
+DAPO	29.38	37.02	44.68	52.08	58.67	63.33
+DAPO w/ UGA	31.15	39.52	47.17	53.37	58.09	63.33
Δ	+1.77	+2.51	+2.48	+1.29	-0.58	0.00

Table 8: Pass@k accuracy (%) on AIME24. Δ denotes performance gaps compared to the baseline without UGA. Base-model scores use Math-Verify.

Model	pass@1	pass@2	pass@4	pass@8	pass@16	pass@32
Qwen3-1.7B-Base	1.88	3.55	6.39	10.61	15.73	20.00
+GRPO	8.02	11.42	14.94	19.55	26.40	33.33
+GRPO w/ UGA	9.38	13.53	17.19	19.56	21.72	23.33
Δ	+1.35	+2.12	+2.26	+0.01	-4.68	-10.00
+DAPO	8.33	12.19	17.36	23.40	29.33	33.33
+DAPO w/ UGA	9.69	14.39	19.50	24.28	28.24	33.33
Δ	+1.35	+2.20	+2.14	+0.88	-1.10	0.00
Qwen3-4B-Base	8.44	13.90	19.89	24.56	29.19	36.67
+GRPO	20.21	26.28	33.12	40.91	47.84	53.33
+GRPO w/ UGA	22.81	29.67	36.50	42.66	48.79	56.67
Δ	+2.60	+3.39	+3.38	+1.75	+0.95	+3.33
+DAPO	25.52	31.47	37.40	43.05	47.25	50.00
+DAPO w/ UGA	26.98	32.98	39.26	45.14	50.74	56.67
Δ	+1.46	+1.52	+1.86	+2.09	+3.49	+6.67

Table 9: Pass@k accuracy (%) on AIME25. Δ denotes performance gaps compared to the baseline without UGA. Base-model scores use Math-Verify.

F Process-Error Diagnostic

Section 2.3 reports an association between trajectory entropy $\mathcal{H}(\tau)$ and process errors on verifier-accepted solutions from Qwen3-1.7B-Base. This appendix records the protocol and a ProcessBench calibration of the critic.

F.1 Evaluation Cohort

We reuse held-out evaluation rollouts that were already generated for the mathematical benchmarks in Section 4.1; trajectories are not regenerated for this analysis. We keep only traces whose automatic answer verifier marks the final answer correct. After excluding 12 traces whose answer or process status the judge marked uncertain, the primary base-model set contains 314 traces from 41 questions, of which 290 are from AMC23. Entropy uses the same top-k=50 token-level estimator as training (Eq. 5). Quintiles are assigned within each benchmark before judging.

F.2 Blinded Judging Protocol

A blinded LLM judge (Grok 4.6) receives the problem, the ground-truth answer, and a numbered candidate solution. The prompt does not include entropy, model name, or the hypothesis under test. The judge assigns one of: valid reasoning, minor omission, major process error, unsupported jump or guess, or uncertain. Only major process errors and unsupported jumps or guesses count as process-invalid; minor omissions are grouped with valid traces in the stacked composition plot. Labels are a single vote. We treat the analysis as an enrichment test rather than an error detector, and we do not interpret absolute rates as judge-independent.

F.3 Critic Calibration

We score the same Grok 4.6 critic on the official 3,400-example ProcessBench dump (Zheng et al., 2025) with a first-error rubric. Official F1 is the harmonic mean of exact first-error-step accuracy (gold-error cases) and all-correct accuracy (predict "no error"), macro-averaged over GSM8K, MATH, OlympiadBench, and Omni-MATH. Table 10 reports a macro F1 of 0.847. Because Section 2.3 only needs a binary validity call, we also score whether the critic predicts any process error. Among the 1,700 ProcessBench cases whose final answer is already correct—the setting that matches the target audit—binary F1 is 0.862 (recall 0.958, precision 0.783; 95% bootstrap CI $[0.840, 0.883]$). The critic is slightly over-eager on this subset (138 false positives versus 22 missed

process errors), so absolute invalid rates in the main text remain judge-dependent even after calibration.

Split	N	Error acc.	Correct acc.	F1
GSM8K	400	0.845	0.990	0.912
MATH	1000	0.862	0.929	0.894
OlympiadBench	1000	0.787	0.847	0.816
Omni-MATH	1000	0.764	0.772	0.768
Macro	3400	–	–	0.847

Table 10: ProcessBench calibration of the Grok 4.6 critic used in Section 2.3. Official F1 is the harmonic mean of exact first-error-step accuracy and all-correct accuracy. Macro is the unweighted mean of the four split F1s.

F.4 Annotated Examples

The two traces below are complete, verifier-accepted Qwen3-1.7B-Base solutions from AMC23. Red boldface marks the false or unsupported step. Example A already errs when summing the fifth row, then continues with a false closed form. Example B produces a longer chain of thought and then jumps to the key quantity without computing it.

F.4.1 Example A: Major Process Error

Verifier-accepted AMC23 solution whose units digit (5) is correct. The earliest error is the fifth-row sum; a false closed form follows. The trace is in the highest within-benchmark entropy quintile ($\mathcal{H}(\tau) = 0.35$).

Question

Rows 1, 2, 3, 4, and 5 of a triangular array of integers are shown below.

1
1 1
1 3 1
1 5 5 1
1 7 11 7 1

Each row after the first row is formed by placing a 1 at each end of the row, and each interior entry is 1 greater than the sum of the two numbers diagonally above it in the previous row. What is the units digits of the sum of the 2023 numbers in the 2023rd row?

Model output

Let's solve this step by step.
First, let's understand the pattern of the triangular array. We have:

1
1 1
1 3 1
1 5 5 1
1 7 11 7 1

From the above pattern, we can observe that the number of entries in each row increases by 1 as we move down the rows. So, the 2023rd row will have 2023 entries.
Now, let's find the sum of the 2023 numbers in the 2023rd row. We can use the pattern we observed to find the sum of the numbers in each row.
For the 1st row, the sum is 1.
For the 2nd row, the sum is $1 + 1 = 2$.
For the 3rd row, the sum is $1 + 3 + 1 = 5$.
For the 4th row, the sum is $1 + 5 + 5 + 1 = 12$.
For the 5th row, the sum is $1 + 7 + 11 + 7 + 1 = 37$.
We can see that the sum of the numbers in each row follows a pattern. The sum of the numbers in the nth row is given by the formula:

Model output *(continued)*

Sum $= 2^{n-1} + 1$
Using this formula, we can find the sum of the numbers in the 2023rd row:
Sum $= 2^{2022} + 1$
Now, we need to find the units digit of the sum. To do this, we can find the units digit of 2^{2022} and add 1.
The units digit of powers of 2 follows a pattern: $2, 4, 8, 6, 2, 4, 8, 6, \ldots$
Since the pattern repeats every 4 powers, we can find the units digit of 2^{2022} by finding the remainder when 2022 is divided by 4.
$2022 \bmod 4 = 2$
So, the units digit of 2^{2022} is the same as the units digit of 2^2, which is 4.
Now, we add 1 to the units digit of 2^{2022}:
$4 + 1 = 5$
Therefore, the units digit of the sum of the 2023 numbers in the 2023rd row is 5.
Answer: 5

Why it is a major process error. The displayed fifth-row entries are $1, 7, 11, 7, 1$, so the sum is 27, not 37. The subsequent closed form $\text{Sum} = 2^{n-1} + 1$ is also false: it yields 17 for $n = 5$, matching neither the true sum nor the model's 37. The true recurrence is $S_n = 2S_{n-1} + n - 2$, hence $S_n = 2^n - n$; the units digit of $2^{2023} - 2023$ happens to be 5 as well. The final answer is therefore correct for the wrong reason.

F.4.2 Example B: Unsupported Jump or Guess

Verifier-accepted AMC23 solution with a longer chain of thought that never computes the key quantity. The trace is also in the highest entropy quintile ($\mathcal{H}(\tau) = 0.37$).

Question

In the state of Coinland, coins have values 6, 10, and 15 cents. Suppose x is the value in cents of the most expensive item in Coinland that cannot be purchased using these coins with exact change. What is the sum of the digits of x?

Model output

To solve this problem, we need to find the largest value x in cents that cannot be formed using the coins of values 6, 10, and 15 cents with exact change. This problem is related to the concept of the Frobenius coin problem, which deals with finding the largest monetary amount that cannot be obtained using any combination of coins of specified denominations.
Step 1: Understanding the Frobenius Coin Problem
The Frobenius number $g(a, b)$ for two coprime integers a and b is given by:

$$g(a, b) = ab - a - b$$

However, in this problem, we have three coins: 6, 10, and 15 cents. To find the largest value that cannot be formed using these coins, we need to consider the combination of these coins.
Step 2: Finding the Largest Value That Cannot Be Formed
We can use the concept of the Frobenius number for multiple coins. The largest value x that cannot be formed using the coins 6, 10, and 15 cents is known as the Frobenius number for these coins.
Step 3: Calculating the Frobenius Number
The Frobenius number for three coins a, b, c can be found using the following formula:

$$g(a, b, c) = abc - ab - ac - bc + a + b + c$$

However, this formula is complex, and it's often easier to find the largest value that cannot be formed by trial and error or by using known results from number theory.
Step 4: Finding the Largest Value x
After some calculations or by using known results, we find that the largest value x that cannot be formed using the coins 6, 10, and 15 cents is 29 cents.
Step 5: Calculating the Sum of the Digits of x
The digits of 29 are 2 and 9. Therefore, the sum of the digits is:

$$2 + 9 = 11$$

Final Answer
Answer: 11

Why it is an unsupported jump. The two-coin Frobenius formula does not apply (the denominations are three and not pairwise coprime), and the three-coin expression in Step 3 is never evaluated. Step 4 then asserts $x = 29$ with no enumeration, generating function, or covering argument. The subsequent digit sum is valid *given* $x = 29$, so the defect is the jump to 29 rather than an arithmetic slip. Outcome-only RL would still assign this trajectory a positive advantage.

G THE USE OF AI ASSISTANTS

AI assistants supported language revision, manuscript organization, and LaTeX typesetting. Grok 4.6 served as the blinded critic for process-error annotation and ProcessBench calibration, as detailed in Appendix F. These labels are treated as judge-dependent diagnostic evidence. The authors are responsible for the manuscript's scientific claims and final content.