FOCUS ON THE CRITICAL: MULTI-GRANULARITY MODULATED CREDIT ASSIGNMENT FOR LONG-HORIZON LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Efficient credit assignment is critical for reinforcement learning (RL) of large language model (LLM) agents in long-horizon interactive environments, where supervision is typically sparse and delayed. However, group-based policy optimization methods often apply a trajectory-level signal uniformly across steps and tokens, which can obscure a small number of pivotal decisions that determine task success. To address this limitation, we propose a *Multi-Granularity Modulated* (MGM) credit assignment framework. It refines the learning signal by integrating token-level Top-K entropy filtering to construct step-level uncertainty estimates, step-level temporal modulation to differentiate early exploration from late-stage execution, and task-level weighting to prioritize challenging queries. Through this unified hierarchical design, MGM effectively highlights pivotal actions, enabling more accurate and efficient policy optimization. Extensive experiments on ALFWorld, WebShop, and Search-Augmented QA demonstrate that our method consistently outperforms state-of-the-art baselines, up to 25.0% absolute improvement based on GRPO and 14.5% based on DAPO. Moreover, we observe a robust unimodal sensitivity to the Top-K ratio, with best performance typically achieved when retaining roughly 80–90% of tokens for entropy estimation, suggesting that excluding a small fraction of low-entropy tokens can improve the signal-to-noise ratio in long-horizon credit assignment.

1 INTRODUCTION

Autonomous agents Achiam et al. (2023); Qwen et al. (2025); Touvron et al. (2023); Liu et al. (2024); Team et al. (2023) are shifting Large Language Models (LLMs) from static question-answering to active, goal-driven problem solving. A critical hallmark of true autonomy is the ability to master long-horizon interactive tasks, such as embodied control (ALFWorld Shridhar et al. (2020)) and web navigation (WebShop Yao et al. (2022a)) Mialon et al. (2023); Schick et al. (2023). However, learning in such dynamic environments relies on sparse, delayed feedback, making credit assignment—identifying which intermediate steps dictate the outcome—a foundational challenge Sutton & Barto (1998); Pignatelli et al. (2023). Reinforcement Learning (RL) is widely adopted as a standard solution. Recent group-based trajectory optimization methods like GRPO (Group-Relative Policy Optimization) Shao et al. (2024) and DAPO (Decoupled Clip and Dynamic Sampling Policy Optimization) Yu et al. (2025) have gained significant traction for their stability and critic-free efficiency.

Despite their success, these group-based estimators often overlook the nuances of fine-grained credit assignment by simply applying a uniform trajectory-level reward to every step within a generation. This uniform treatment dilutes the reinforcement signal, failing to distinguish the handful of pivotal decisions that determine task success from trivial executions or syntactic bridging. Therefore, we argue that explicitly isolating and emphasizing these decisive actions from the noisy reasoning trajectory is the key to unlocking highly effective policy optimization in long-horizon environments.

To address this, existing works have employed Process Reward Models (PRMs) Lightman et al. (2023) to provide dense, step-level supervision signals. While effective in improving credit assignment, PRMs typically require costly human annotation of intermediate steps, limiting their scalabil-

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

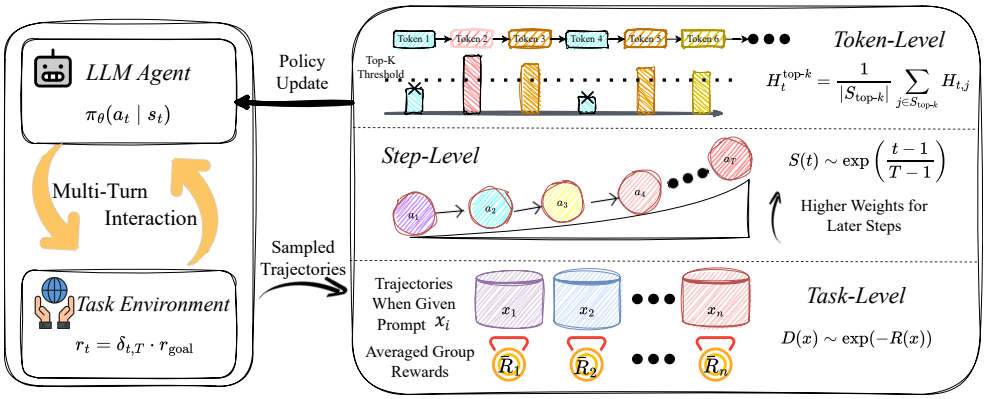


Figure 1: Overview of our **Multi-Granularity Modulated (MGM)** framework. The framework uses signals at three granularities to construct step-wise advantages: (1) **Token-Level**: Top-K entropy filtering aggregates high-entropy token information into a step-level uncertainty estimate; (2) **Step-Level**: A temporal modulation curve ($S(t)$) assigns progressively higher relative weights to later steps; (3) **Task-Level**: A difficulty-aware modulator ($D(x)$) prioritizes challenging queries with lower average rewards. These components transform coarse trajectory-level rewards into step-wise learning signals.

ity. As an alternative, recent works have explored using intrinsic signals, such as policy entropy, to modulate step-wise advantages Wang et al. (2025a). While high step-wise entropy generally signals high uncertainty, existing methods Wang et al. (2025a) typically calculate it by simply averaging over all tokens in a step, thereby treating syntactic fillers and critical decision tokens equally. We argue that this overlooks the skewed (80/20) nature of LLM generation Wang et al. (2025b), where only a small subset of pivotal tokens truly determine the semantic outcome. When computing step-level uncertainty via standard averaging, functional tokens with low entropy act as syntactic noise, artificially deflating the entropy score and masking the true semantic uncertainty of critical reasoning steps. Current methods lack a mechanism to disentangle this syntactic redundancy from semantic decision-making.

In this paper, we propose an RLVR framework that integrates signals at three different levels to assign credit for long-horizon multi-turn tasks, as illustrated in Figure 1. First and foremost is the **Token-Level**, where we introduce a Top-K Entropy filtering mechanism. Instead of averaging all tokens, we estimate step uncertainty using only the top-$k\%$ highest-entropy tokens. This acts as a semantic denoiser, robustly isolating significant branching points from syntactic fillers. Complementing this, we operate at the **Step-Level** by applying a normalized temporal schedule that assigns relatively larger weights to later steps, and at the **Task-Level** by dynamically upweighting harder tasks to prevent overfitting to easy samples during training.

Our contributions are summarized as follows:

- **Analytical Insights:** We theoretically show that entropy-based filtering provides an effective approximation to true semantic uncertainty by suppressing low-entropy syntactic variations that do not contribute to meaningful decision signals. This analysis further explains the unimodal behavior of Top-K entropy estimation: retaining too few tokens loses semantic evidence, while retaining too many reintroduces syntactic noise.
- **Methodological Innovation:** We propose the Multi-Granularity Modulated (MGM) framework, which synergistically integrates token-level Top-K entropy filtering, step-level temporal scheduling, and task-level difficulty weighting for efficient fine-grained credit assignment.
- **Empirical Performance:** MGM consistently improves the state-of-the-art group-based RL baselines across extensive benchmarks, including ALFWorld, WebShop, and Search-Augmented QA, delivering up to a 25.0% absolute improvement over GRPO and 14.5% over DAPO.

2 Related Work

RL Optimization for LLMs. Traditional RLHF via PPO Schulman et al. (2017) is often heavy and unstable due to its reliance on a critic network. While preference methods like DPO Rafailov et al. (2023); Azar et al. (2024); Ahmadian et al. (2024) bypass reward models entirely, group-based trajectory optimization methods (e.g., GRPO Shao et al. (2024), DAPO Yu et al. (2025), and GiGPO Feng et al. (2025)) have emerged as highly efficient, critic-free alternatives for RL. We build upon these robust group-based estimators. However, recognizing their coarse-grained nature, we enhance their core advantage estimation mechanism via multi-granularity modulations to address their inherent limitation in distinguishing fine-grained step contributions.

Credit Assignment and Process Supervision. Granular credit assignment in sparse-reward RL remains a persistent challenge. Explicit strategies like Process Reward Models (PRMs) Lightman et al. (2023) offer dense feedback but require costly step-level human annotations. Test-time strategies such as MCTS Yao et al. (2023) or self-correction Shinn et al. (2023); Madaan et al. (2023) bypass annotations but introduce significant inference overhead. In contrast, our approach achieves implicit credit assignment entirely during training. By leveraging the model's intrinsic probability distribution to dynamically isolate pivotal steps, we negate the need for external reward models or costly inference-time search.

Entropy and Uncertainty in LLMs. Entropy is a prevalent proxy for LLM uncertainty Manakul et al. (2023); Kadavath et al. (2022); Diao et al. (2024). In RL, EMPG Wang et al. (2025a) mitigates "Entropy-Gradient Coupling" by leveraging step-level policy uncertainty for variance reduction. Concurrently, an "80/20 rule" Wang et al. (2025b) reveals that a tiny subset of high-entropy tokens dictates generation quality in static math reasoning. Translating these insights to noisy, interactive environments is non-trivial: existing work Wang et al. (2025a) evaluating step uncertainty by uniformly averaging all tokens inadvertently incorporates syntactic fillers alongside critical semantic decisions. We bridge this gap through a Top-K entropy mechanism, strategically filtering out low-entropy distractor tokens to strictly capture true decision uncertainty for step-wise modulation.

3 Methodology

3.1 Preliminaries

Setup. We formulate interactive LLM generation as a sparse-reward Markov Decision Process (MDP). Given a query x, the agent generates a trajectory $\tau = (s_1, a_1, \ldots, s_T, a_T)$, receiving a scalar reward $r(\tau)$ only at the terminal step. To optimize the policy efficiently without a critic network, we build upon GRPO Shao et al. (2024). For a group of G trajectories, GRPO computes a baseline-normalized advantage $A_i = (r_i - \text{mean}(\mathbf{r}))/\text{std}(\mathbf{r})$ and maximizes the surrogate objective:

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left(\mathcal{L}_{i,t}^{CLIP}(\theta) - \beta D_{KL}\right)\right]. \tag{1}$$

Crucially, GRPO applies the trajectory-level scalar advantage A_i *uniformly* across all tokens and steps.

Autoregressive Policy Entropy. The token-level prediction uncertainty of LLMs $w_{t,j}^{(i)}$ within step t is defined by its standard Shannon entropy:

$$H_{t,j}^{(i)} = -\sum_{v\in\mathcal{V}} p_\theta(v \mid w_{t,<j}^{(i)}) \log p_\theta(v \mid w_{t,<j}^{(i)}). \tag{2}$$

To represent the confidence of a multi-token action a_t, a conventional approach is naive averaging: $H_{naive}(a_t) = \frac{1}{|a_t|}\sum_j H_{t,j}^{(i)}$. However, this uniformly averages high-frequency, low-entropy syntactic tokens (e.g., punctuation) with critical high-entropy decision tokens, artificially masking the true semantic uncertainty and motivating our more granular estimation.

3.2 Theoretical Motivation: Semantic Denoising for Entropy-Modulated Credit Assignment

The core limitation of group-relative optimization under sparse terminal rewards is not merely that the reward is delayed, but that the resulting scalar advantage is broadcast to a sequence of heterogeneous generation events. Some tokens determine a semantic branch of the action, whereas many others only realize a syntactic template. This distinction is theoretically important because policy uncertainty is intrinsically coupled with gradient magnitude. For a categorical policy parameterized by logits, the expected squared score-function norm satisfies

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\|\nabla_{\mathbf{z}}\log\pi_\theta(a\mid s)\|_2^2\right]=1-\exp\left(-H_2(\pi_\theta(\cdot\mid s))\right), \tag{3}$$

where H_2 is the Rényi entropy of order 2 (details in B.2). Thus, high-uncertainty decisions naturally induce larger policy-gradient updates even before any explicit credit modulation is applied. Entropy-based modulation is therefore a principled way to stabilize sparse-reward training, but only if the entropy used for modulation faithfully estimates *decision uncertainty* rather than low-level surface-form uncertainty.

We formalize this point with a separated token-mixture view of an action step. Let $h_{t,j}=H_{t,j}^{(i)}$ denote the entropy of the j-th token in step t. Suppose the token positions of an action a_t can be decomposed into a semantic decision set $\mathcal{P}_t$ and a syntactic set $\mathcal{N}_t$, with $q_t=|\mathcal{P}_t|$ and $m_t=|a_t|$. Define the oracle semantic uncertainty as

$$U_t=\frac{1}{q_t}\sum_{j\in\mathcal{P}_t}h_{t,j}. \tag{4}$$

The conventional all-token entropy $\widehat{H}_t^{\mathrm{all}}$ mixes U_t with low-entropy syntactic tokens. In contrast, our Top-K estimator $\widehat{H}_t^K$ averages only the $r_t=\lceil Km_t\rceil$ largest token entropies.

Lemma 3.1 (Existence of an Effective Top-K Semantic Denoiser)**.** *Semantic decision tokens exhibit higher average uncertainty than the overall step average (i.e., $U_t\geq\widehat{H}_t^{\mathrm{all}}$). Then, there necessarily exists a target retention ratio $K\in(0,1]$ where the Top-K entropy estimator $\widehat{H}_t^K$ satisfies:*

$$\widehat{H}_t^{\mathrm{all}}\leq\widehat{H}_t^K\leq U_t. \tag{5}$$

Consequently, evaluating step entropy at this optimal K yields a provably tighter approximation to the true semantic decision uncertainty:

$$0\leq U_t-\widehat{H}_t^K\leq U_t-\widehat{H}_t^{\mathrm{all}}. \tag{6}$$

Lemma 3.1 explains why naively averaging all tokens can systematically underestimate the uncertainty of pivotal action choices: adding low-entropy function words, delimiters, and formatting tokens contracts the estimate toward syntactic certainty. Top-K entropy is therefore not a heuristic truncation, but a denoised estimator of the uncertainty that is actually coupled with meaningful policy updates.

We next connect this denoising effect to MGM's advantage modulation. Let $g(\cdot)$ be the decreasing entropy-scaling function used in Section 3.3, and let $B_t=D(x)S(t)$ denote the product of the task-level and step-level positive priors. For any step-entropy estimator $\widehat{H}_t$, define the mean-normalized positive modulation factor

$$M_t(\widehat{H})=\frac{B_t g(\widehat{H}_t)}{\mathrm{Mean}_{\mathrm{batch}}(B_t\cdot g(\widehat{H}_t))}. \tag{7}$$

Theorem 3.2 (Stable semantic credit allocation)**.** *Let $M_t^\star=M_t(U)$ be the oracle modulation factor that observes semantic uncertainty U_t. Suppose $\log g$ is L-Lipschitz on the entropy range of the batch. By Lemma 3.1,*

$$\max_t\left|\log M_t(\widehat{H}^K)-\log M_t^\star\right|\leq 2L\max_t|U_t-\widehat{H}_t^K|\leq 2L\max_t|U_t-\widehat{H}_t^{\mathrm{all}}|. \tag{8}$$

Theorem 3.2 shows that MGM's Top-K entropy produces a multiplicative credit allocation that is closer to the oracle semantic allocation than all-token entropy, while the mean normalization prevents the modulation from simply increasing the global learning rate. Combined with the entropy–gradient coupling above, this provides the theoretical motivation for the three-level design of MGM: Top-K entropy supplies a high-fidelity semantic uncertainty signal, $g(\cdot)$ counteracts uncontrolled entropy-induced gradient amplification, $S(t)$ introduces a normalized position-based prior, and $D(x)$ reallocates the same update budget toward harder tasks.

3.3 Method Design

High-entropy token-based step entropy calculation. To more accurately capture the semantic role of step entropy, we propose a refined calculation method that focuses on high-entropy tokens within each step, improving upon the naive averaging discussed in Section 3.1. Specifically, for a given step t in trajectory τ_i, we utilize the token-level entropy $H_{t,j}^{(i)}$ defined in Eq. 2.

We sort these token entropies in descending order and select the top $k\%$ as high-entropy tokens. The step entropy is then computed using a selection indicator function $I(j \in S_t^{(i)})$ that identifies whether token position j belongs to the top-k high-entropy set $S_t^{(i)}$:

$$H_t^{(i)} = \frac{1}{\sum_{j=1}^{m} I(j \in S_t^{(i)})} \sum_{j=1}^{m} I(j \in S_t^{(i)}) \cdot H_{t,j}^{(i)}, \tag{9}$$

where $I(j \in S_t^{(i)}) = 1$ if $j \in S_t^{(i)}$, and 0 otherwise. By filtering out low-entropy syntactic tokens, $H_t^{(i)}$ serves as a more sensitive proxy for the model's confidence in key reasoning decisions. This token-level selection changes the step-level entropy estimate, not the allocation of credit among tokens within a step: all tokens in the same step share its modulated advantage.

Step index modulation for dynamic credit assignment. Recognizing that different steps along a trajectory may contribute unequally to the final reward, we introduce a step index modulation factor to dynamically adjust the credit assigned to each step. Unlike traditional temporal discounting (γ-decay), which applies a monotonically vanishing weight to early steps and causes severe numerical instability in long-horizon reasoning (further analyzed in Appendix E.4), our approach fixes $\gamma = 1$ to broadcast the terminal reward uniformly, and directly acts on the gradient weights via an exponentially-scaled, mean-normalized temporal index. In the context of multi-turn interactive tasks where the trajectory length $T > 1$ (e.g., the maximum interaction turns is 50 in ALFWorld), we normalize the step index for each time step $t \in \{1, 2, \ldots, T\}$ as $t_{\text{norm}} = \frac{t-1}{T-1}$, and then apply mean normalization to obtain the step index modulation factor:

$$S(t) = \frac{T \cdot \exp(t_{\text{norm}})}{\sum_{t'=1}^{T} \exp(t'_{\text{norm}})}. \tag{10}$$

The factor $S(t)$ is a position-based prior, not an estimate of causal contribution. Its smaller but nonzero early-step weights temper the temporal component of updates to earlier actions, which may allow greater tolerance for exploration while retaining a learning signal; later steps receive larger relative weights. We evaluate the usefulness of this inductive bias empirically.

Task difficulty modulation for conquering challenging tasks. Once the model has acquired basic task competence in the later stage of training, we introduce a task difficulty modulation mechanism to encourage the model to tackle more challenging tasks. Specifically, for a set of trajectories $T_x = \{\tau_1, \tau_2, \ldots, \tau_N\}$ generated from the same query $x \in \mathcal{D}$, we compute the cumulative reward $R(\tau_i)$ for each trajectory. The average group reward

$$R(x) = \frac{1}{N} \sum_{\tau_i \in T_x} R(\tau_i), \tag{11}$$

is used to estimate the difficulty of the task. We then normalize the task difficulty and apply mean normalization to obtain the task difficulty modulation factor:

$$R_{\text{norm}}(x) = \frac{R(x) - \min_{x' \in \mathcal{D}} R(x')}{\max_{x' \in \mathcal{D}} R(x') - \min_{x' \in \mathcal{D}} R(x')}, \tag{12}$$

Table 1: Performance comparison on ALFWorld and WebShop. For ALFWorld, we report the average success rate (%) for each subtask as well as the overall result. For WebShop, we report both the average score and the average success rate (%).

Method	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	All	Task Score	Success Rate
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
Prompting Qwen2.5-1.5B-Instruct	5.9	5.5	3.3	9.7	4.2	0.0	4.1	23.1	5.2
Prompting ReAct	17.4	20.5	15.7	6.2	7.7	2.0	12.8	40.1	11.3
Prompting Reflexion	35.3	22.2	21.7	13.6	19.4	3.7	21.8	55.8	21.9
RL Training PPO (with critic)	64.8	40.5	57.1	60.6	46.4	47.4	54.4	73.8	51.5
RL Training GiGPO	94.4	67.5	94.8	94.4	79.8	76.4	86.7	83.1	65.0
RL Training GRPO	87.9	40.0	78.1	35.7	65.2	44.4	65.6	78.0	58.2
with EMPG	85.5	33.5	78.9	76.2	74.7	69.1	73.7	80.4	60.8
with Ours	97.2	91.7	84.2	81.8	95.5	85.7	90.6	83.3	63.3
RL Training DAPO	88.1	61.4	82.5	90.1	83.9	69.5	80.8	85.9	73.2
with EMPG	97.7	80.7	87.5	87.0	88.3	80.0	88.1	86.8	73.8
with Ours	100.0	100.0	100.0	90.9	96.2	80.0	95.3	90.4	78.9
Prompting Qwen2.5-7B-Instruct	33.4	21.6	19.3	6.9	2.8	3.2	14.8	26.4	7.8
Prompting ReAct	48.5	35.4	34.3	13.2	18.2	17.6	31.2	46.2	19.5
Prompting Reflexion	62.0	41.6	44.9	30.9	36.3	23.8	42.7	58.1	28.8
RL Training PPO (with critic)	92.3	64.0	92.5	89.5	80.3	68.8	80.4	81.4	68.7
RL Training GiGPO	97.7	82.7	98.8	83.7	89.3	79.2	90.8	84.4	72.8
RL Training GRPO	88.8	43.7	88.1	70.3	77.7	56.8	74.8	77.8	65.6
with EMPG	92.9	75.2	74.8	86.3	73.7	65.3	78.5	81.0	69.3
with Ours	100.0	91.7	100.0	91.7	79.0	93.3	93.8	81.7	73.1
RL Training DAPO	98.9	86.1	94.9	83.2	81.4	90.1	90.0	90.6	79.6
with EMPG	99.0	86.8	97.3	94.9	75.8	90.3	91.6	92.0	82.7
with Ours	100.0	91.0	94.6	91.7	95.6	94.7	95.3	93.4	84.0

$$D(x) = \frac{|\mathcal{D}| \cdot \exp(-R_{\text{norm}}(x))}{\sum_{x' \in \mathcal{D}} \exp(-R_{\text{norm}}(x'))}. \tag{13}$$

In our experiments, we find that task difficulty modulation is consistently beneficial compared to the baseline, yielding the most significant gains in the later stages of training (see details in Appendix E.2). When the model has acquired some ability to solve the current tasks, the modulation factor $D(x)$ assigns higher weights to more challenging tasks, encouraging the model to focus on overcoming these tasks.

Overall advantage modulation. Combining the components discussed above, we define the overall modulated step-wise advantage for each time step t in trajectory τ_i as follows. First, we compute the mean-normalized modulation factor $M_t(H^{(i)})$ using the task difficulty factor $D(x)$, the step index modulation factor $S(t)$, and the self-calibrating function $g(H_t^{(i)})$, normalized by their batch mean. Then, we assemble the preliminary advantage $A_{\text{pre}}(i,t)$ by multiplying this factor with the group-relative advantage $A^{(i)}$ and adding the future clarity bonus $\zeta \cdot f(H_{t+1}^{(i)})$:

$$A_{\text{pre}}(i,t) = A^{(i)} \cdot M_t(H^{(i)}) + \zeta \cdot f(H_{t+1}^{(i)}). \tag{14}$$

Finally, the final step-wise advantage $A_{\text{final}}(i,t)$ is obtained by subtracting the batch mean of A_{pre} to zero-center all step-wise advantages. The detailed formulations are provided in Appendix C.

4 Experiments and Analysis

4.1 Experiment Setup

Benchmarks. We evaluate our method across three diverse domains requiring long-horizon reasoning with sparse rewards. *ALFWorld* Shridhar et al. (2020) simulates household environments, requiring agents to execute multi-step textual actions to achieve goals across six task categories . *WebShop* Yao et al. (2022a) offers a realistic e-commerce setting where agents must navigate web pages and process HTML observations to purchase products matching user instructions. Additionally, we further assess tool-use ability on *Search-Augmented QA* tasks Feng et al. (2025), including single-hop

Table 2: Comparison on Single-Hop and Multi-Hop QA datasets on Qwen2.5-3B-Instruct. The superscript † indicates datasets where we train, while ⋆ indicates datasets used for zero-shot evaluation. To ensure a fair comparison, we strictly adhere to the experimental settings and hyperparameters defined in the GiGPO scripts.

Method	Single-Hop QA			Multi-Hop QA				Avg.
	NQ†	TriviaQA⋆	PopQA⋆	HotpotQA†	2Wiki⋆	MuSiQue⋆	Bamboogle⋆	
R1-Instruct	27.0	53.7	19.9	23.7	29.2	7.2	29.3	27.1
Search-R1	34.1	54.5	37.8	32.4	31.9	10.3	26.4	32.5
ZeroSearch	41.4	57.4	44.8	27.4	30.0	9.8	11.1	31.7
StepSearch	-	-	-	34.5	32.0	17.4	34.4	-
GiGPO	42.0	59.5	42.4	36.9	37.0	12.6	64.1	42.1
Ours	44.9	61.1	47.0	39.3	40.4	14.8	67.7	45.9

datasets (NQ Kwiatkowski et al. (2019), TriviaQA Joshi et al. (2017), PopQA Mallen et al. (2023)) and multi-hop datasets (HotpotQA Yang et al. (2018), 2Wiki Ho et al. (2020), MuSiQue Trivedi et al. (2022), Bamboogle Press et al. (2023)), where agents must actively retrieve and synthesize information via search engine queries for multi-turn reasoning.

Baselines. For ALFWorld and WebShop, we compare our approach against a diverse set of strong baselines using Qwen2.5-Instruct (1.5B and 7B) as the backbone. We first report results for closed-source models, GPT-4o Achiam et al. (2023) and Gemini-2.5-Pro Team et al. (2023), to establish performance reference points. We then compare with prompting-based agents, including ReAct Yao et al. (2022b), and Reflexion Shinn et al. (2023), which leverage in-context learning without parameter updates. Finally, we compare with reinforcement learning fine-tuning approaches: PPO Schulman et al. (2017) as a representative actor-critic method, and critic-free baselines including GiGPO Feng et al. (2025), GRPO Shao et al. (2024), and DAPO Yu et al. (2025). For Search-Augmented QA tasks, we evaluate against R1-Instruct, Search-R1 Jin et al. (2025), ZeroSearch Sun et al. (2025), StepSearch Wang et al. (2025c) and GiGPO Feng et al. (2025), using Qwen2.5-Instruct (3B)

Implementation Details. Our implementation is built upon the GiGPO public codebase. Since our proposed modulation framework is designed to enhance group-based optimization, we integrate it with both GRPO and DAPO, using them as the primary baselines for direct comparison. We employ Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct as backbone models for ALFWorld and WebShop benchmarks, while utilizing Qwen2.5-3B-Instruct for the Search-Augmented QA tasks. To ensure a fair comparison, we strictly adhere to the hyperparameter configurations reported in the baseline methods. Comprehensive training details and settings are provided in the Appendix E.

4.2 Main Results

Results on ALFWorld and WebShop. Tables 1 and 2 summarize the main results. On ALFWorld, our modulation consistently boosts performance across model scales. With the Qwen2.5-1.5B backbone, integrating our method with GRPO increases the "All" score significantly from 65.6% to 90.6%. Combined with DAPO, it raises the score from 80.8% (DAPO baseline) to 95.3% (DAPO + Ours). A similar trend is observed with the Qwen2.5-7B backbone: GRPO improves substantially from 74.8% to 93.8%, and DAPO improves from 90.0% to 95.3%. WebShop results further validate our approach. In the 1.5B setting, DAPO + Ours achieves a Task Score of 90.4 and Success Rate of 78.9, surpassing the baseline's 85.9 and 73.2. For 7B, we also see consistent gains, with Success Rate improving from 79.6 (DAPO) to 84.0 (DAPO + Ours). These results indicate that our multi-granularity modulation consistently strengthens group-based optimizers across benchmarks and model scales. We additionally verify the cross-model generalization of our framework on the Llama-3.2-3B-Instruct architecture, with detailed results provided in Appendix E.6.

Results on Search-Augmented QA Tasks. Table 2 reports results on Search-Augmented QA tasks using Qwen2.5-3B. When building upon GRPO, our method achieves an average success rate of 45.9, outperforming the GiGPO baseline (42.1). Substantial improvements are observed in challenging multi-hop reasoning tasks, such as HotpotQA (36.9 $\rightarrow$ 39.3) and Bamboogle (64.1 $\rightarrow$ 67.7), as well as in single-hop tasks like PopQA (42.4 $\rightarrow$ 47.0). Crucially, unlike ALFWorld where trajectories often span $\geq$ 25 steps, these QA tasks have a much shorter horizon (max 4 steps). Although

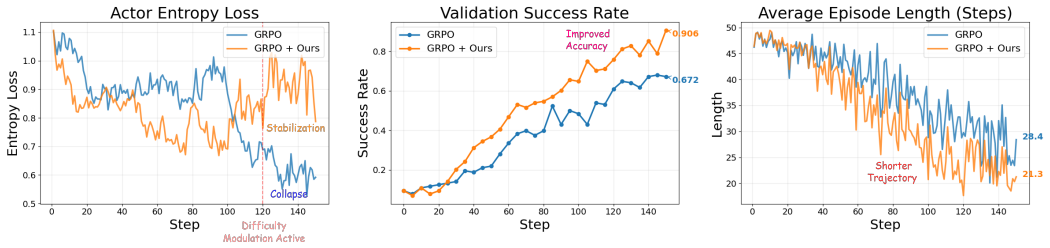


Figure 2: Training dynamics on ALFWorld (Qwen2.5-1.5B). (Left) The **Actor Entropy Loss** curve demonstrates that our Difficulty-Based Modulation (activated at step 120) effectively mitigates entropy collapse, maintaining higher policy diversity. (Middle) This sustained exploration translates to *Higher Success*, achieving a superior convergence point (90.6% vs 67.2%). (Right) The **Average Episode Length** confirms that our method is *More Efficient*, requiring significantly fewer steps to complete tasks compared to the baseline.

our multi-granularity modulation is primarily designed to enhance long-chain reasoning, it maintains robust performance even in such short-term scenarios. This demonstrates the versatility of our approach, proving that refined credit assignment is beneficial across varying trajectory lengths, not just for extensive multi-turn interactions.

4.3 Training Dynamics and Efficiency Analysis

Stabilized Entropy via Difficulty Modulation. We analyze the training stability by visualizing the actor entropy loss (Figure 2, Left). Standard GRPO exhibits entropy decay, often leading to premature convergence. In contrast, our approach activates **Difficulty-Based Modulation** at step 120, a phase where the model has already acquired basic task capabilities. By assigning higher weights to more challenging instances, this mechanism incentivizes the agent to conquer distinct hard tasks rather than overfitting to easy ones. This effectively mitigates entropy collapse and encourages the generation of **diverse trajectories**, ensuring the policy explores broad solution spaces to maximize performance on complex problems.

Success and Episode Length. Figure 2 (Middle & Right) shows that MGM achieves a higher success rate while using fewer steps on average. The position-based schedule gives greater relative weight to later steps while retaining a signal for early exploration. These observations are consistent with the full MGM design, but the plot alone does not isolate the contribution of step-index modulation to episode length.

4.4 Ablation Studies

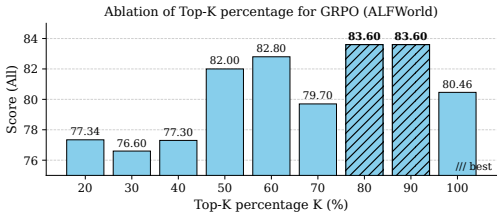


Figure 3: Ablation of Top-K percentages for GRPO on ALFWorld. Performance exhibits a robust unimodal trend, validating that filtering low-entropy tokens effectively isolates semantic decisions. The optimal region lies around $K \approx 80\%$–90%, whereas full-token averaging degrades accuracy due to syntactic noise.

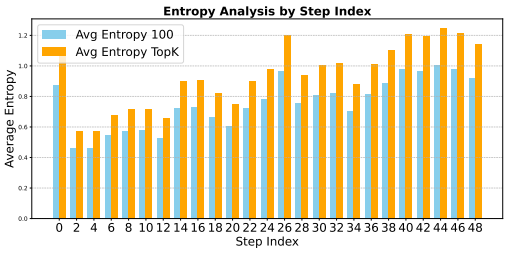


Figure 4: Step-wise entropy analysis on ALFWorld. It contrasts full-token averaging (**Avg Entropy 100**) with our Top-K modulation (**Avg Entropy TopK**). Our method acts as a *semantic denoiser*, filtering low-entropy syntactic noise to amplify the intrinsic uncertainty signal while preserving the **distributional topology**.

Table 3: Ablation study on modulation components on ALFWorld on Qwen2.5-1.5B-Instruct. We incrementally add components to the DAPO baseline to validate their effectiveness.

Method	Pick	Look	Clean	Heat	Cool	Pick2	All
DAPO (Baseline)	88.1	61.4	82.5	90.1	83.9	69.5	80.8
+ Top-K Entropy	96.4	78.6	100.0	82.4	88.9	80.0	89.1
+ Step-wise Mod.	100.0	96.9	100.0	100.0	85.7	86.7	93.0
+ Difficulty Mod.	100.0	83.3	94.7	100.0	95.5	82.1	93.0
Full Method (Ours)	100.0	100.0	100.0	90.9	96.2	80.0	95.3

Ablation on Modulation Components. To dissect the contributions of each modulation component, we conduct an incremental study on ALFWorld (Table 3) using Qwen2.5-1.5B-Instruct. Starting from the DAPO baseline (80.8%), introducing High-Entropy Token-Based modulation alone yields a substantial gain to 89.1%. Adding either Step Index or Task Difficulty modulation further boosts performance to 93.0%. Finally, the full framework achieves the highest success rate of 95.3%, confirming that token, step, and task-level signals are mutually complementary and collectively essential for optimal long-horizon reasoning. To formally isolate these independent effects and prove their orthogonality, we present a complete factorial breakdown ablation in Appendix E.5.

Ablation on Top-K Percentage. We investigate sensitivity to the Top-K percentage on ALFWorld in isolation (i.e., with step and difficulty modulations disabled) (Figure 3). The results exhibit a clear **unimodal trend**. Performance is suboptimal at low K (20–40%) due to aggressive filtering but rises significantly to peak at $K = 80$–90%. Crucially, reverting to full-token averaging ($K = 100\%$) causes a performance drop. This confirms that the bottom 10–20% of tokens act as syntactic noise that dilutes the learning signal. Filtering this fraction achieves an optimal signal-to-noise ratio, maximizing semantic coverage without noise contamination. This empirical unimodality directly corroborates Lemma 3.1, which guarantees the existence of an optimal $K^\star$ where the Top-K estimator most closely approximates the oracle semantic uncertainty: retaining too few tokens loses decision evidence, while retaining too many reintroduces syntactic noise. To further validate the cross-domain universality of this linguistic prior, we provide an extended sensitivity analysis on the WebShop benchmark in Appendix E.3.

4.5 Analysis of Step-wise Entropy Representation

Recent work EMPG Wang et al. (2025a) has demonstrated that naively applying the mathematical reasoning-derived "80/20 rule" to multi-turn interactive tasks can be detrimental to training. Motivated by this, and to validate our High-Entropy Token-Based method, we analyze step-wise entropy on *ALFWorld* using 2,000 trajectories generated by the same trained policy. Figure 4 compares standard all-token averaging (Avg Entropy 100) with our Top-K entropy (Avg Entropy TopK).

Semantic Denoising without Distortion. Although the average of the top-$k\%$ values is necessarily higher than the overall mean, the key issue is what is excluded. All-token averaging is diluted by low-entropy functional tokens, such as structural markers, punctuation, and articles, which are useful for syntax but carry little decision information. Top-K entropy filters out this syntactic redundancy, yielding a higher and more faithful estimate of decision uncertainty. At the same time, the peaks and troughs of the Top-K curve remain aligned with those of the standard curve, showing that the method amplifies the intrinsic uncertainty signal while preserving its distributional topology and avoiding distortion.

5 Conclusion

In this paper, we focus on the limitations of coarse-grained credit assignment in group-based trajectory optimization for LLM agents. To mitigate this issue, we propose a **Multi-Granularity Modulated** framework that synergizes Token-level Top-K semantic denoising, Step-level temporal modulation, and Task-level difficulty modulation. Empirical results across ALFWorld, WebShop, and Search-Augmented QA demonstrate that our method consistently outperforms state-of-the-art baselines. Furthermore, our validation of the "80/20 rule" in long-horizon reasoning highlights the critical importance of filtering syntactic noise, offering a scalable and efficient pathway for aligning agents in sparse-reward environments without reliance on dense human annotations. One limitation of our study is that the effectiveness of our framework on ultra-large-scale models and more diverse real-world environments remains to be fully explored. A promising direction for future work involves extending our framework to multi-modal agent environments.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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. *arXiv preprint arXiv:2402.14740*, 2024.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

Shizhe Diao, Pengcheng Wang, Yong Lin, Rui Pan, Xiang Liu, and Tong Zhang. Active prompting with chain-of-thought for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1330–1350, 2024.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*, 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.

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.

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.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

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.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.

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, 2023.

Potsawee Manakul, Adian Liusie, and Mark Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 9004–9017, 2023.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, et al. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*, 2023.

Eduardo Pignatelli, Johan Ferret, Matthieu Geist, Thomas Mesnard, Hado van Hasselt, Olivier Pietquin, and Laura Toni. A survey of temporal credit assignment in deep reinforcement learning. *arXiv preprint arXiv:2312.01072*, 2023.

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.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

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.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 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.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

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.

Richard S. Sutton and Andrew G. Barto. Reinforcement learning: An introduction. *IEEE Trans. Neural Networks*, 9:1054–1054, 1998. URL https://api.semanticscholar.org/CorpusID:60035920.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multi-hop questions via single-hop question composition, 2022. URL https://arxiv.org/abs/2108.00573.

Jiawei Wang, Jiacai Liu, Yuqian Fu, Yingru Li, Xintao Wang, Yuan Lin, Yu Yue, Lin Zhang, Yang Wang, and Ke Wang. Harnessing uncertainty: Entropy-modulated policy gradients for long-horizon llm agents, 2025a. URL https://arxiv.org/abs/2509.09265.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025b.

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*, 2025c.

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.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022b.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

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.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in llm reasoning. *arXiv preprint arXiv:2506.01347*, 2025.

A GRPO Objective Details

The full objective function of GRPO, including the proximal clipping mechanism, is defined as:

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}_{q\sim P(Q),\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left(\min\left(\rho_{i,t}A_i, \text{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)A_i\right) - \beta\mathbb{D}_{KL}(\pi_\theta||\pi_{\text{ref}})\right)\right] \quad (15)$$

where $\rho_{i,t} = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}$ is the importance sampling ratio. This ratio measures how much the new policy deviates from the old one. The term $\text{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)$ constrains this ratio within a trusted region (typically $\epsilon = 0.2$) to prevent excessively large updates that could destabilize training. The $\min$ operator then selects the pessimistic bound between the unclipped and clipped objectives, ensuring safe and monotonic policy improvement.

B Theoretical Analysis of Semantic Entropy Modulation

In this section, we provide the full derivation for the theoretical motivation in Section 3.2. The analysis has two parts. First, following EMPG Wang et al. (2025a), we recall that policy entropy is quantitatively coupled with the magnitude of score-function gradients. Second, we prove that the proposed Top-K entropy estimator is closer to the oracle semantic uncertainty than all-token averaging under low-entropy syntactic contamination, and that this improvement translates into a more stable mean-normalized credit allocation.

B.1 Entropy–Gradient Coupling

Lemma B.1. *Consider a categorical policy $\pi_\theta(\cdot \mid s)$ parameterized by logits $\mathbf{z} \in \mathbb{R}^{|\mathcal{V}|}$, where $\pi_\theta(a \mid s) = \frac{\exp(z_a)}{\sum_{v\in\mathcal{V}}\exp(z_v)}$. Let $\mathbf{g}_a = \nabla_{\mathbf{z}} \log \pi_\theta(a \mid s)$ denote the score function with respect to the logits for a specific action a. Then*

$$\|\mathbf{g}_a\|_2^2 = 1 - 2\pi_\theta(a \mid s) + \sum_{v\in\mathcal{V}} \pi_\theta(v \mid s)^2. \quad (16)$$

Proof. For any vocabulary element v, the derivative of the log-probability is

$$\frac{\partial \log \pi_\theta(a \mid s)}{\partial z_v} = \mathbb{I}(v = a) - \pi_\theta(v \mid s). \quad (17)$$

Therefore,

$$\|\nabla_{\mathbf{z}} \log \pi_\theta(a \mid s)\|_2^2 = \sum_{v\in\mathcal{V}} \left(\mathbb{I}(v = a) - \pi_\theta(v \mid s)\right)^2 \quad (18)$$

$$= \left(1 - \pi_\theta(a \mid s)\right)^2 + \sum_{v\neq a} \pi_\theta(v \mid s)^2 \quad (19)$$

$$= 1 - 2\pi_\theta(a \mid s) + \sum_{v\in\mathcal{V}} \pi_\theta(v \mid s)^2. \quad (20)$$

□

Proposition B.2. *The expected squared score-function norm over actions sampled from $\pi_\theta(\cdot \mid s)$ is*

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\|\mathbf{g}_a\|_2^2\right] = 1 - \exp\left(-H_2(\pi_\theta(\cdot \mid s))\right), \quad (21)$$

where $H_2(\pi) = -\log\sum_{v\in\mathcal{V}} \pi(v)^2$ is the Rényi entropy of order 2.

Proof. Taking the expectation of Lemma B.1 with respect to $a \sim \pi_\theta(\cdot \mid s)$ gives

$$\mathbb{E}_{a\sim\pi}\left[\|\mathbf{g}_a\|_2^2\right] = \sum_{a\in\mathcal{V}} \pi(a)\left(1 - 2\pi(a) + \sum_{v\in\mathcal{V}} \pi(v)^2\right) \quad (22)$$

$$= \sum_a \pi(a) - 2\sum_a \pi(a)^2 + \left(\sum_a \pi(a)\right)\left(\sum_v \pi(v)^2\right) \quad (23)$$

$$= 1 - \sum_{v\in\mathcal{V}} \pi(v)^2. \quad (24)$$

By the definition of Rényi-2 entropy,

$$\sum_{v\in\mathcal{V}} \pi(v)^2 = \exp(-H_2(\pi)). \tag{25}$$

Substituting this identity yields the result. □

Proposition B.2 shows that more diffuse policy distributions have larger expected score-function norms. Since the Shannon entropy H_1 upper-bounds H_2 and both quantify distributional spread, Shannon token entropy provides a practical proxy for locating steps where gradient magnitudes can become large. MGM uses this signal through a decreasing scaling function $g(\cdot)$, but the following derivation shows that the signal should be computed from high-entropy decision tokens rather than all generated tokens.

B.2 Proof of Top-K Semantic Denoising

Proof of Lemma 3.1. Let m_t be the total number of tokens in step t, and sort the token entropies in descending order $h_{(1)} \geq h_{(2)} \geq \cdots \geq h_{(m_t)}$. For $r \in [1, m_t]$, define

$$A_r = \frac{1}{r}\sum_{j=1}^{r} h_{(j)}. \tag{26}$$

Then $\widehat{H}_t^K = A_{r_t}$ with $r_t = \lceil K m_t \rceil$.

Monotonicity. For any $r < m_t$,

$$A_{r+1} = \frac{rA_r + h_{(r+1)}}{r+1} \leq A_r, \tag{27}$$

since $h_{(r+1)} \leq h_{(r)} \leq A_r$. Hence A_r is non-increasing in r, and in particular $A_{m_t} \leq A_r$ for all r. This yields $\widehat{H}_t^{\text{all}} = A_{m_t} \leq \widehat{H}_t^K$.

Existence. Because U_t is an average of a subset of the m_t entropies, $U_t \leq \max_j h_{t,j} = h_{(1)} = A_1$. Together with $U_t \geq A_{m_t}$, we have

$$A_{m_t} \leq U_t \leq A_1. \tag{28}$$

Define the nonempty set $\mathcal{R} = \{r \in [1, m_t] : A_r \leq U_t\}$ and let $r^\star = \min \mathcal{R}$. Then $A_{r^\star} \leq U_t$ and (if $r^\star > 1$) $A_{r^\star - 1} \geq U_t$. Setting $K^\star = r^\star / m_t$ gives $r_t = \lceil K^\star m_t \rceil = r^\star$ and therefore

$$\widehat{H}_t^{\text{all}} \leq \widehat{H}_t^{K^\star} = A_{r^\star} \leq U_t. \tag{29}$$

Subtracting from U_t yields

$$0 \leq U_t - \widehat{H}_t^{K^\star} \leq U_t - \widehat{H}_t^{\text{all}}. \tag{30}$$

This inequality formally demonstrates that the Top-K estimator systematically reduces the estimation gap caused by syntactic dilution, yielding a strictly more accurate approximation of the oracle semantic uncertainty to guide the advantage modulation. □

B.3 Proof of Stable Mean-Normalized Credit Allocation

Proof of Theorem 3.2. For a positive prior $B_t = D(x)S(t)$ and an entropy estimator $\widehat{H}_t$, let T denote the total number of steps across all trajectories in the current optimization batch. Then define

$$M_t(\widehat{H}) = \frac{B_t g(\widehat{H}_t)}{\frac{1}{T}\sum_{s=1}^{T} B_s g(\widehat{H}_s)}. \tag{31}$$

The mean-preservation property is immediate:

$$\frac{1}{T}\sum_{t=1}^{T} M_t(\widehat{H}) = \frac{\frac{1}{T}\sum_{t=1}^{T} B_t g(\widehat{H}_t)}{\frac{1}{T}\sum_{s=1}^{T} B_s g(\widehat{H}_s)} = 1. \tag{32}$$

Thus, the modulation reallocates a fixed update budget over steps instead of changing the trajectory-level learning rate.

Let $M_t^\star = M_t(U)$ be the oracle semantic modulator and define

$$\delta_t = \log g(\widehat{H}_t) - \log g(U_t). \tag{33}$$

Since $\log g$ is L-Lipschitz,

$$|\delta_t| \le L|\widehat{H}_t - U_t|. \tag{34}$$

Let $C_t^\star = B_t g(U_t) > 0$. Then

$$\log \frac{M_t(\widehat{H})}{M_t^\star} = \log \frac{B_t g(\widehat{H}_t)}{B_t g(U_t)} - \log \frac{\sum_{s=1}^T B_s g(\widehat{H}_s)}{\sum_{s=1}^T B_s g(U_s)} \tag{35}$$

$$= \delta_t - \log \left(\frac{\sum_{s=1}^T C_s^\star \exp(\delta_s)}{\sum_{s=1}^T C_s^\star} \right). \tag{36}$$

The second term is the logarithm of a weighted average of $\exp(\delta_s)$ with positive weights $C_s^\star$. Hence it lies between $\min_s \delta_s$ and $\max_s \delta_s$. Therefore,

$$\left| \log \frac{M_t(\widehat{H})}{M_t^\star} \right| \le \max_s \delta_s - \min_s \delta_s \le 2 \max_s |\delta_s| \le 2L \max_s |\widehat{H}_s - U_s|. \tag{37}$$

Applying this bound with $\widehat{H} = \widehat{H}^K$ and then using Lemma 3.1 yields

$$\max_t \left| \log M_t(\widehat{H}^K) - \log M_t^\star \right| \le 2L \max_t |U_t - \widehat{H}_t^K| \le 2L \max_t |U_t - \widehat{H}_t^{\text{all}}|. \tag{38}$$

This proves that Top-K entropy produces a mean-normalized modulation closer to the oracle semantic credit allocation than all-token averaging. □

C Details of Modulation Functions

In the overall advantage modulation (Section 3.3), we adopt the self-calibrating gradient scaling function $g(\cdot)$ and the future clarity bonus function $f(\cdot)$ from EMPG Wang et al. (2025a).

Self-Calibrating Gradient Scaling $g(H_t^{(i)})$. The function $g(\cdot)$ aims to attenuate the natural coupling between entropy and policy gradients. It adopts a Softmax-like normalization form to dynamically scale the advantage based on step entropy:

$$g(H_t^{(i)}) = \frac{\exp(-k \cdot H_{\text{norm},t}^{(i)})}{\frac{1}{\sum_{j=1}^{N_B} T_j} \sum_{j=1}^{N_B} \sum_{t'=1}^{T_j} \exp\left(-k \cdot H_{\text{norm},t'}^{(i)}\right)} \tag{39}$$

where N_B is the number of trajectories in the batch, T_j is the length of the j-th trajectory, and $H_{\text{norm},t}^{(i)}$ is the normalized step entropy defined by Min-Max normalization within the current batch:

$$H_{\text{norm},t}^{(i)} = \frac{H_t^{(i)} - H_{\min}}{H_{\max} - H_{\min} + \epsilon} \tag{40}$$

where $H_{\min}$ and $H_{\max}$ are the minimum and maximum step entropy values observed in the current batch, respectively, and ϵ is a small constant for numerical stability. This formulation ensures that steps with higher entropy (higher uncertainty) receive relatively lower weights in the gradient update, enhancing training stability.

Future Clarity Bonus $f(H_{t+1}^{(i)})$. The future clarity bonus encourages the model to take actions that lead to more certain future states. It is defined as:

$$f(H_{t+1}^{(i)}) = \exp(-k' \cdot H_{\text{norm},t+1}^{(i)}) \tag{41}$$

where k' is a scaling hyperparameter. A lower entropy at the next step $H_{t+1}^{(i)}$ results in a larger bonus, rewarding the policy for reducing future uncertainty. For the terminal step $t = T$, no successor state exists and the bonus is set to zero (i.e., $f(H_{T+1}^{(i)}) = 0$).

D PSEUDO CODE OF MGM

We present the pseudo-code for the key components of our Multi-Granularity Modulated (MGM) framework. Algorithm 1 describes the calculation of step entropy based on top-K high-entropy tokens. Algorithm 2 and Algorithm 3 detail the computation of step index modulation and task difficulty modulation factors, respectively. Finally, Algorithm 4 illustrates the overall advantage modulation process.

Algorithm 1 Top-K Entropy-Based Step Entropy Calculation

Input: Token-level entropies $\mathbf{H}_{token} = \{H_{t,j}^{(i)}\}_{j=1}^{m}$ for step t, Top-K ratio k_{ratio}
Output: Step entropy $H_t^{(i)}$
$n \leftarrow \text{length}(\mathbf{H}_{token})$
$n_{top} \leftarrow \max(1, \lceil k_{ratio} \cdot n \rceil)$
$\mathbf{H}_{top} \leftarrow \text{TopK}(\mathbf{H}_{token}, n_{top})$ {Select top n_{top} values}
$H_t^{(i)} \leftarrow \text{Mean}(\mathbf{H}_{top})$
Return $H_t^{(i)}$

Algorithm 2 Step Index Modulation

Input: Batch of trajectories $\mathcal{T}$, scaling factor k_{step}
Output: Step modulation factors $\{S(t)\}$
for each trajectory $\tau \in \mathcal{T}$ **do**
  $T \leftarrow \text{Length}(\tau)$
  for $t = 1$ to T **do**
    $t_{norm} \leftarrow \frac{t-1}{T-1}$
    $S_{raw}(t) \leftarrow \exp(k_{step} \cdot t_{norm})$
  end for
  $S_{mean} \leftarrow \frac{1}{T} \sum_{t=1}^{T} S_{raw}(t)$
  for $t = 1$ to T **do**
    $S(t) \leftarrow S_{raw}(t)/S_{mean}$
  end for
end for
Return $\{S(t)\}$

Algorithm 3 Task Difficulty Modulation

Input: Dataset $\mathcal{D}$, episode rewards R, scaling factor β
Output: Task difficulty factors $\{D(x)\}$
Initialize dictionary $AvgRewards$
for each unique query $x \in \mathcal{D}$ **do**
  Find all trajectories $\mathcal{T}_x$ associated with x
  $AvgRewards[x] \leftarrow \text{Mean}(\{R(\tau) \mid \tau \in \mathcal{T}_x\})$
end for
$R_{min} \leftarrow \min(AvgRewards.values())$
$R_{max} \leftarrow \max(AvgRewards.values())$
for each $x \in \mathcal{D}$ **do**
  $R_{norm} \leftarrow \frac{AvgRewards[x] - R_{min}}{R_{max} - R_{min} + \epsilon}$
  $D_{raw}(x) \leftarrow \exp(-\beta \cdot R_{norm})$
end for
$D_{mean} \leftarrow \text{Mean}(D_{raw}.values())$
for each $x \in \mathcal{D}$ **do**
  $D(x) \leftarrow D_{raw}(x)/D_{mean}$
end for
Return $\{D(x)\}$

Algorithm 4 Overall MGM Advantage Computation

Input: Batch $\mathcal{B}$, hyperparameters k, k', ζ
Output: Modulated advantages A_{mgm}
Extract step entropies $H_t^{(i)}$ using Alg. 1
Normalize step entropies to $H_{norm,t}^{(i)}$ via Min-Max in batch
Compute Step Modulation $S(t)$ using Alg. 2
Compute Task Difficulty $D(x)$ using Alg. 3 if enabled
for each trajectory τ_i in $\mathcal{B}$ **do**
  $T \leftarrow \text{Length}(\tau_i)$
  for each step t in $1 \dots T$ **do**
    $g(H_t^{(i)}) \leftarrow \exp(-k \cdot H_{norm,t}^{(i)})$
    $W_{i,t} \leftarrow g(H_t^{(i)}) \cdot S(t) \cdot D(x)$
  end for
end for
$W_{mean} \leftarrow \text{Mean}(\{W_{i,t} \mid \forall i, t\})$
for each trajectory τ_i in $\mathcal{B}$ **do**
  $T \leftarrow \text{Length}(\tau_i)$
  for each step t in $1 \dots T$ **do**
    $M_t(H^{(i)}) \leftarrow W_{i,t}/W_{mean}$
    $f(H_{t+1}^{(i)}) \leftarrow \exp(-k' \cdot H_{norm,t+1}^{(i)})$
    $A_{pre}(i,t) \leftarrow A^{(i)} \cdot M_t(H^{(i)}) + \zeta \cdot f(H_{t+1}^{(i)})$
  end for
end for
$A_{mean} \leftarrow \text{Mean}(\{A_{pre}(i,t) \mid \forall i, t\})$
for each trajectory τ_i in $\mathcal{B}$ **do**
  $T \leftarrow \text{Length}(\tau_i)$
  for each step t in $1 \dots T$ **do**
    $A_{mgm}(i,t) \leftarrow A_{pre}(i,t) - A_{mean}$
  end for
end for
Return A_{mgm}

E Details of Experiment

E.1 Training Details

We provide the detailed hyperparameters used for our experiments in this section to ensure reproducibility. Table 4 lists the key hyperparameters for the long-horizon interactive benchmarks, ALFWorld and WebShop. Both tasks share a majority of the configuration settings, including learning rates and penalty coefficients, demonstrating the robustness of our method. The experiments were conducted on NVIDIA A100 GPUs.

Table 5 details the specific configurations for the Search-Augmented QA tasks. These tasks involve shorter horizons but require precise search actions. Accordingly, we adjusted parameters such as group size and mini-batch size. We utilized a rule-based reward system where successful answers receive a reward of 1, failures 0, and invalid search actions are penalized with -0.01 to encourage valid tool usage.

E.2 Sensitivity Analysis on When to Start Difficulty Modulation

We investigate the impact of the starting step for Difficulty Modulation (T_{start}) on the ALFWorld benchmark using the Qwen2.5-1.5B-Instruct model. The total training steps are fixed at 150. As shown in Table 6, the method's performance is sensitive to the timing of this module's introduction.

The results reveal a non-monotonic relationship regardless of the general trend. Introducing the modulation too early (Step 0, 82.03%) yields limited gains, likely because the difficulty estimation

Table 4: Key Hyperparameters for ALFWorld and WebShop Experiments. Parameters marked with * are specific to our method, while others remain consistent with baselines to ensure fair comparison.

Parameter	**Value**
Actor Learning Rate	1e-6
KL Loss Coefficient	0.01
KL Penalty	low var kl
Entropy Coefficient	0.001
Clip High (DAPO)	0.28
Clip Low (DAPO)	0.2
Clip Low/High (GRPO)	0.2
Batch Size	16
Training Step	150
Rollout Group Size	8
Rollout Temperature	1.0
ζ	0.05
k, k'	1.0
Max Actions (ALFWorld)	50
Max Actions (WebShop)	15
History Observation	2
GPUs*	$4 \times$ RTX 6000 Ada (1.5B), $8 \times$ A100 (7B)
Difficulty Modulation Start Step*	120
Top-K Entropy Ratio*	80%

Table 5: Hyperparameters for Search-Augmented QA Experiments. Parameters marked with * are specific to our method, while others remain consistent with baselines to ensure fair comparison.

Parameter	**Value**
Max Prompt Length	4096
Max Response Length	512
Max Turn	4
Actor Learning Rate	1e-6
Reward (Success / Failure)	1 / 0
Reward (Invalid Action)	-0.01
Train Data Size	256
Rollout Group Size	5
Rollout Temperature	1.0
Validation Temperature	0.0
Mini-batch Size	512
KL Coefficient	0.001
GPUs*	$4 \times$ A100
Difficulty Modulation Start Step*	120
Top-K Entropy Ratio*	80%

is noisy when the policy is unstable. Interestingly, we observe a strong local optimum at Step 30 (87.50%), followed by a performance dip in the intermediate phase (Steps 60-90). This sensitivity suggests that during the main exploration phase, aggressively re-weighting tasks based on difficulty might interfere with the policy's ability to learn foundational skills.

The global optimum is achieved at Step 120 (90.62%), where the modulation acts as a *finishing move* in the final 20% of training. This timing allows the model to first converge on easier tasks and then focus its remaining capacity on conquering the harder tail of the distribution. The drop at Step 150 (81.25%) represents the baseline performance without effective modulation, confirming the module's necessity. Therefore, we recommend treating Difficulty Modulation as a curriculum refinement tool to be enabled in the later stages of training.

Table 6: Sensitivity Analysis of Difficulty Modulation Start Step on ALFWorld (Qwen2.5-1.5B-Instruct). Total training steps are 150. The method achieves peak performance when modulation is introduced in the final stage (Step 120).

Start Step	0	30	60	90	120	150
Success Rate (%)	82.03	87.50	83.59	83.59	**90.62**	81.25

E.3 Cross-Domain Robustness of the Top-K Threshold

A potential concern with introducing the Top-K entropy threshold is its generalization across different applications. We emphasize that the optimal range $K \in [80\%, 90\%]$ is not a brittle empirical trick requiring heavy tuning, but a fundamental linguistic prior in LLMs. Across varied LLM tasks, roughly 10–20% of tokens typically constitute "syntactic glue" (e.g., punctuation, formatting, and stop words). As our Top-K threshold acts as a stable semantic denoiser, consistently dropping this 10–20% of uniformly low-entropy grammar noise isolates true decision uncertainty regardless of the specific task.

To empirically prove this cross-domain robustness, we extend the sensitivity analysis of the Top-K parameter to the **WebShop** benchmark (HTML navigation). Despite WebShop having vastly different action spaces (e.g., `click[...]`, `search[...]`) and token distributions compared to the embodied physical tasks of ALFWorld, the optimal range aligns identically, as shown in Table 7.

Table 7: Cross-Domain Sensitivity Analysis of the Top-K Ratio on ALFWorld and WebShop. The optimal performance on both benchmarks identically peaks around $K = 80\% - 90\%$, demonstrating that the $10 - 20\%$ syntactic noise reduction is a robust cross-domain prior.

Top-K Ratio	50%	60%	70%	80%	90%	100% (Naive)
ALFWorld (Succ. Rate %)	82.0	82.8	79.7	**83.6**	**83.6**	80.5
WebShop (Task Score)	76.5	77.9	80.8	83.3	**83.4**	81.6
WebShop (Succ. Rate %)	57.8	53.5	59.0	63.3	**63.7**	60.2

In both domains, performance degrades only when K is too aggressive ($< 70\%$, which begins destroying semantic coverage) or absent (100%, which admits non-semantic syntax noise). This outcome confirms that our parameterization exploits a robust property of language modeling, enabling cross-domain universality without the need for extensive hyperparameter sweeps.

E.4 Distinction Between Step-Level Modulation and Gamma Decay

While both our step-level modulation and traditional γ-decay share a philosophical connection as temporal weighting functions, they diverge fundamentally in their mathematical consequences. Standard discounting defines returns as $R_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k}$. In sparse terminal reward settings, this forces returns to simplify to $R_t = \gamma^{T-t} r_T$. For any $\gamma < 1$, the credit assigned to non-terminal steps is exponentially decayed backwards in time. This aggressively shrinks the credit signal and introduces severe numerical vanishing problems, functionally obliterating early exploratory signals.

In contrast, our step-level modulation fixes $\gamma = 1$ (broadcasting the raw reward uniformly) and instead adjusts the *gradient weights* via an exponentially-scaled, mean-normalized temporal index. This ensures the trajectory's mean weight remains ~ 1.0, safely bounding the variance for early exploration while explicitly prioritizing determining final steps.

To illustrate this concretely on a 50-step trajectory:

- With a typical $\gamma = 0.95$, the first step's weight is severely vanished: $\gamma^{49} \approx 0.081$ (while the last step receives 1.0).
- With our Step-Level Modulation, the first step is gracefully bounded: $50 \cdot e^0 / \sum_{k=0}^{49} e^{\frac{k}{49}} \approx 0.58$ (while the last step is amplified to ≈ 1.6).

As shown, our method maintains a dynamically stable variance bound, safely preserving early exploration paths while prioritizing the final acts, whereas γ-decay functionally obliterates early signals. To further validate this empirically, we conducted two ablations.

First, we tested a discounted baseline ($\gamma = 0.95$) against standard DAPO ($\gamma = 1$) and our method on WebShop using the Qwen2.5-1.5B-Instruct model. As shown in Table 8, applying γ-decay functionally worsens performance compared to $\gamma = 1$, confirming that exponential vanishing is a structural flaw in long-horizon reasoning.

Table 8: Ablation of Discount Factor (γ) on WebShop (Qwen2.5-1.5B-Instruct). Standard γ-decay drastically drops performance compared to the raw baseline computation ($\gamma = 1$).

Setting	DAPO ($\gamma = 1$)	Discounted DAPO ($\gamma = 0.95$)	**Ours**
Task Score (%)	85.9	84.6	**90.4**
Succ. Rate (%)	73.2	66.1	**78.9**

To conclusively demonstrate that simply tuning γ cannot mimic the performance of our framework, we conducted an exhaustive hyperparameter search over $\gamma \in \{0.95, 0.90, \ldots, 0.60\}$ using GRPO on ALFWorld.

Table 9: Systematic sweep of γ-decay values (ALFWorld, GRPO, 105 epochs). Shrinking γ exacerbates signal loss, culminating in premature sub-optimal convergence, severely lagging behind our Multi-Granularity Modulated design.

Setting	Base (1.0)	0.95	0.90	0.85	0.80	0.75	0.70	0.65	0.60	**Ours**
Succ. Rate	52.3	39.8	39.8	39.1	32.8	33.6	35.2	35.2	46.9	**75.0**

As the results in Table 9 demonstrate, none of the tested γ values recovers the performance of the undiscounted baseline or approaches that of the full MGM framework. We also reversed the temporal weighting to favor earlier steps. On ALFWorld (Qwen2.5-1.5B-Instruct), this change reduced performance from 95.3% to 87.5%, supporting the chosen temporal orientation in this setting without implying that later steps are necessarily more causally important.

E.5 Factorial Breakdown Ablation of Modulation Levels

To conclusively verify that our proxies do not induce compounding or cascading errors, and to formally isolate their independent effects, we present a complete Factorial Breakdown Ablation on the ALFWorld benchmark using the Qwen2.5-1.5B-Instruct model (Table 10).

Table 10: Factorial Breakdown Ablation on ALFWorld (Qwen2.5-1.5B-Instruct). The individual modules demonstrate strong independent gains over the DAPO baseline, and their combination yields the maximum performance without overlapping interference.

Method	Pick	Look	Clean	Heat	Cool	Pick2	All
DAPO (Baseline)	88.1	61.4	82.5	90.1	83.9	69.5	80.8
Token-Level Only	96.4	78.6	100.0	82.4	88.9	80.0	89.1
Step-Level Only	100.0	94.1	100.0	85.7	77.8	82.6	91.4
Task-Level Only	92.1	66.7	95.5	100.0	81.8	84.2	88.3
Full Method (Ours)	100.0	100.0	100.0	90.9	96.2	80.0	95.3

The results clearly isolate the gains. Rather than amplifying errors or overlapping redundantly, the mechanisms function synergistically at entirely different dimensions: the Token-level provides clean intra-step signal denoising, the Step-level anchors inter-step temporal variance, and the Task-level balances batch-wise trajectory gradients. Their combination yields structural gains greater than any single proxy alone, confirming their complementarity.

E.6 Cross-Model Generalization

To further test cross-model generalization, we extended full evaluations to the **Llama-3.2-3B-Instruct** architecture on the ALFWorld benchmark. Modern RL optimization literature Zhu et al. (2025); Yue et al. (2025) notes that RL behaviors and inherent efficacy vary significantly across baseline backbone families. Therefore, absolute metrics should be evaluated intra-model, whereas relative gains strongly dictate algorithmic robustness.

Table 11: Cross-Model Generalization Evaluation on ALFWorld. MGM yields consistent positive optimization gains across distinct model architectures.

Optimizer	**Qwen-2.5-1.5B-Instruct**	**Llama-3.2-3B-Instruct**
Baseline (GRPO)	65.6%	58.6%
Ours (Full MGM)	**90.6%**	**61.7%**

As shown in Table 11, MGM yields a 3.1% absolute improvement over standard GRPO on Llama-3.2-3B. Despite the differing margin magnitude relative to Qwen2.5-1.5B, the consistent positive gains confirm the framework's conceptual generalizability and safety against architecture-specific overfitting.

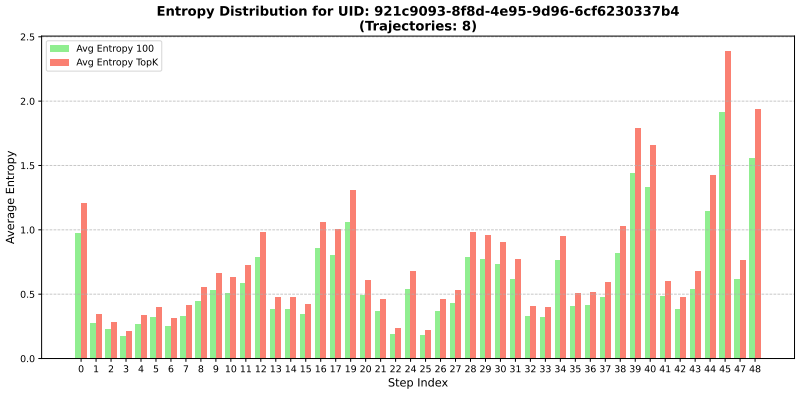
E.7 Analysis of Step-Wise Entropy Distribution Across Different Queries

To validate the generalizability of our granular entropy calculation, we visualize the step-wise entropy distribution of the Qwen2.5-1.5B-Instruct model across three distinct ALFWorld queries in Figure 5. The figure compares the standard average entropy (using all tokens) with our Top-K entropy (filtering for high-entropy tokens) across the steps of generated trajectories. We observe three critical characteristics that support our design:

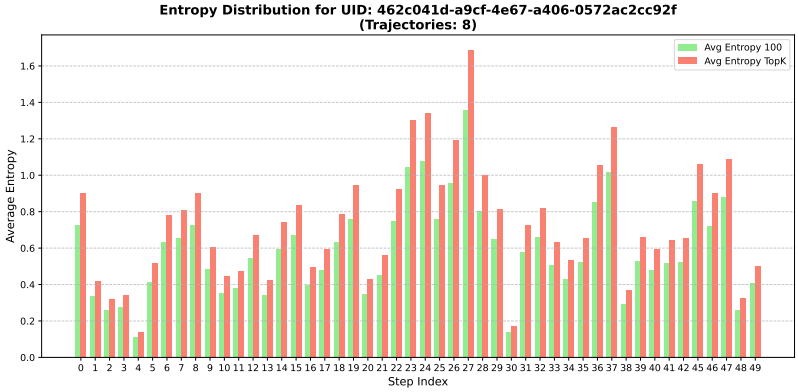
Universal Stochasticity and Non-Monotonicity. The entropy distribution does not follow a simple linear decay or monotonic trend as the trajectory progresses. Instead, it exhibits significant stochasticity, with sharp high-entropy peaks appearing even at late stages. Importantly, this pattern is universal across different queries, confirming that critical decision points—where the model faces high uncertainty—are irregularly distributed throughout the reasoning chain regardless of the specific task content.

Topology Preservation with Signal Amplification. Our Top-K entropy filtering preserves the global topological structure of the original entropy distribution; the peaks and valleys align perfectly between the two metrics across all queries. Crucially, the Top-K metric (Red bars) consistently yields higher values than the Naive Average (Green bars), effectively amplifying the uncertainty signal. This indicates that our method successfully filters out the noise of low-entropy syntactic tokens without distorting the relative difficulty landscape of the trajectory.

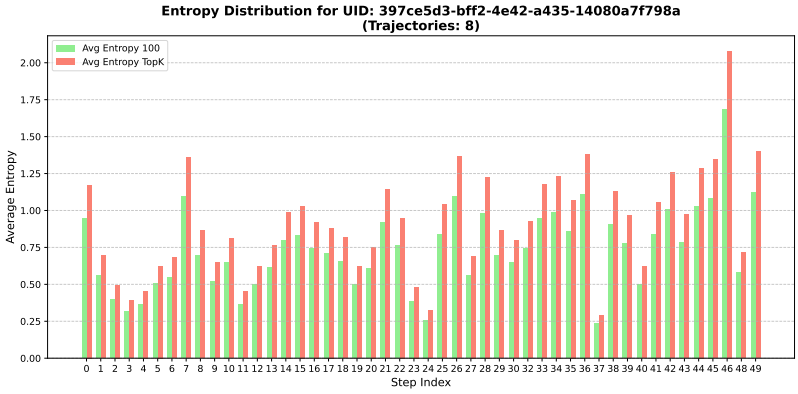
High-Fidelity Semantic Alignment. The Top-K metric exhibits sharper contrast between steps compared to the standard average. By excluding the dampening effect of confident, trivial tokens, our metric provides a higher-fidelity representation of the true semantic uncertainty. This sharper signal allows the modulation mechanism to more precisely distinguish between routine operations and decisive reasoning steps for any given query, enabling more targeted credit assignment.



(a) Entropy Distribution - Query 1



(b) Entropy Distribution - Query 2



(c) Entropy Distribution - Query 3

Figure 5: Step-wise entropy distribution analysis across three distinct queries (a-c). The Green bars represent standard average entropy, while the Red bars represent our Top-K entropy. This visualization confirms that the stochastic and non-monotonic nature of effective uncertainty is a universal phenomenon across specific tasks. In all cases, the Top-K metric successfully amplifies relative peaks while preserving the topological structure, proving its robustness across varying semantic contexts.

F BROADER IMPACT, LIMITATIONS AND SAFEGUARDS

Our experiments are conducted in sandboxed benchmarks rather than live web environments. While improved credit assignment can make agents more efficient, stronger autonomous agents may also increase risks of unintended tool use, web automation misuse, and error amplification in multi-step decision making. These risks are not unique to our method but are amplified by any technique that accelerates agent capability. We therefore recommend deploying such systems with action constraints, logging, human approval for high-impact actions, and domain-specific safety filters. We do not release agents connected to unrestricted real-world tools.

AI USE STATEMENT

We used generative AI tools to assist with language editing, LaTeX formatting, software implementation and debugging, and the development and checking of mathematical proofs during the preparation of this manuscript. We did not use generative AI tools to generate experimental data, fabricate or alter experimental results, or replace the authors' scientific judgment. The authors reviewed and verified all AI-assisted text, code, and proofs, and tested the code. We take full responsibility for the final content of this work, including all text, claims, analyses, code, and other artifacts produced with the aid of generative AI.