Information Geometric Uncertainty Modeling for Agentic Reinforcement Learning

Weikang Zhao*, Jing Hua*, Xiujuan Wang*, Haoyu Wang*, Mengzhen Kang*†‡
*State Key Laboratory of Multimodal Artificial Intelligence Systems
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
†Faculty of Innovation Engineering
University of Science and Technology, Macau 999078, China
{zhaoweikang2023, jing.hua, xiujuan.wang, haoyu.wang, mengzhen.kang}@ia.ac.cn

***Abstract*—Recent agentic reinforcement learning (RL) methods have made strong progress in training tool-using large language models (LLMs). A common approach is to trigger extra rollouts at uncertain decision points, often detected by high entropy. However, entropy is only a scalar measure. It often fails to capture the geometric structure of the predictive distribution, particularly after tool interactions. As a result, entropy struggle to tell informative uncertainty from random noise. To address this, we propose GUA-RL (information Geometric Uncertainty modeling for Agentic Reinforcement Learning), an information geometric framework for adaptive sampling in agentic RL. Instead of relying on entropy, GUA-RL tracks fisher-based geometric signals after each tool interaction. It allocates sampling budget to the decision points that are most informative for exploration. We evaluate GUA-RL on 13 long-horizon reasoning tasks spanning competition-level math, knowledge-intensive multi-hop QA, and deep search. Across these benchmarks, GUA-RL delivers the best overall results and remains consistently effective. Overall, our findings show that information geometric uncertainty provides a practical replacement for entropy-only rollout in tool-augmented agentic reinforcement learning.**

I. Introduction

Large language models are rapidly evolving from single-turn generators into tool-using agents that plan, act, and revise over long horizons. In realistic deployments—mathematical problem solving with verifiers [1], [2], multi-hop question answering with retrieval [3], or web navigation with search—an agent must interleave reasoning with external tool [4], [5], interpret environment feedback, and decide what to do next. This turns tool-use into a sequential decision making process rather than a single-step generation problem. Each action changes the future information the agent will observe and constrains what it can do next, making credit assignment across steps essential. As a result, the agent faces a large and branching search space, where early tool choices can have long-term effects but provide little immediate feedback. This makes naive trial and error inefficient. Therefore, a key challenge in agentic reinforcement learning is exploration: deciding where and when to spend the sampling budget to discover better tool-use strategies.

A practical challenge is that modern LLMs agentic RL pipelines already rely on multiple rollouts per prompt to stabilize optimization and reduce variance (e.g., group-based policy optimization [6], [7]). However, naïvely increasing the number of full trajectories is prohibitively expensive in tool-rich environments. Recent work therefore advocates adaptive sampling: keep a global rollout budget, but allocate additional samples only at critical decision points, especially after tool feedback, where the agent's belief state may change abruptly. Entropy-guided branching is a representative approach in this direction [8]–[10]: it detects "high-uncertainty" regions after tool call and triggers additional sampling to diversify trajectories. Yet the success of this idea hinges on a subtle question: which kind of uncertainty is actually worth branching on?

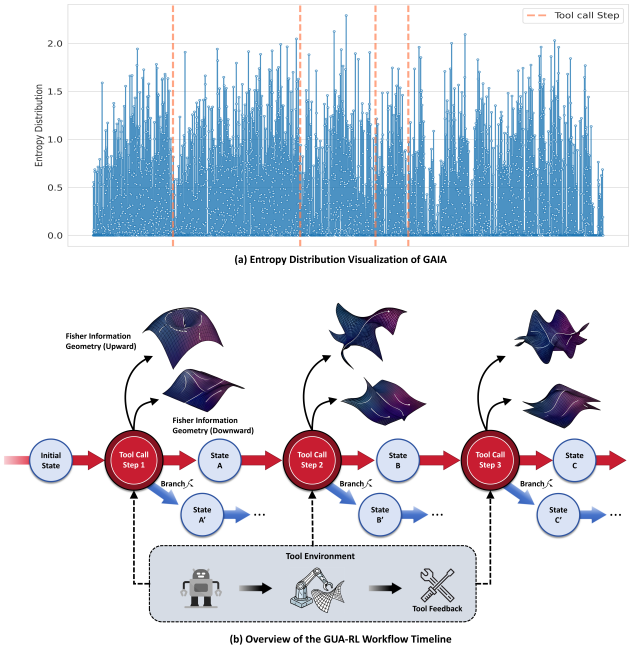


Fig. 1. **(a)**: Token entropy distribution after tool call. **(b)**: GUA-RL Workflow Timeline. At each tool call step, the agent computes the Frobenius norm of the Fisher information matrix to quantify the local curvature of the probability manifold. High-curvature regions (upward) indicate informed uncertainty where the model's output distribution exhibits complex geometric structure, prompting deeper exploration. Low-curvature regions (downward) suggest confident predictions, enabling exploitation of learned policies.

‡ is the corresponding author.

This question exposes a key limitation of entropy as a branching signal. Entropy is a scalar summary of probability dispersion: it can be high when the model is genuinely uncertain among several plausible next steps, but it can also be high when the distribution is nearly uniform due to lack of information or instability. In tool-using settings, these two cases have very different implications. After a tool returns feedback, the agent might enter an "actionable" regime where multiple continuations exist and branching can reveal better plans; alternatively, it might enter an "uninformative" regime where the model is simply confused, and additional sampling mostly wastes budget. Because entropy ignores the distribution's structure, it cannot tell whether high uncertainty reflects several plausible next actions or simple guesswork with no clear direction.

We argue that what matters for branching is not only "how spread" the distribution is, but how the distribution's geometry changes when the environment provides new information. Tool feedback can reshape the agent's predictive distribution in an anisotropic way: some directions like candidate actions or reasoning steps become much more sensitive, while others change little. Such anisotropic changes are exactly what information geometry aims to capture. In particular, the fisher information metric provides the space of token distributions with a Riemannian geometry, measuring how sensitive the model's predictions are to local changes.

Motivated by this, we propose **GUA-RL**(information Geometric Uncertainty modeling for Agentic Reinforcement Learning), an information geometric framework for uncertainty-aware adaptive sampling in agentic RL. As shown on Figure 1(b), instead of triggering branch rollouts based on entropy spikes, GUA-RL uses fisher information geometric signals to decide when and where to branch after each tool interaction. When tool feedback causes a large and structured change in the agent's predictive distribution, the model branches, steering it toward more exploratory behavior.

We evaluate GUA-RL on a broad suite of long-horizon reasoning benchmarks spanning mathematical reasoning, knowledge-intensive multi-hop QA, and deep search tasks, under a standardized training setup and strong agentic RL baselines. Across these settings, fisher geometric guidance consistently improves tool-use exploration compared with entropy-guided branching, highlighting that geometry-aware uncertainty is a more reliable signal for adaptive branching decisions in tool-interactive environments. In summary, our main contributions are:

- **A geometric perspective on branching for tool-using agents.** We view adaptive branching as detecting *meaningful changes* in the model's predictive distribution after tool feedback, and explain why entropy, as a scalar number spread measure, can be unreliable.
- **GUA-RL: Fisher-based adaptive sampling for agentic RL.** We introduce a rollout strategy that uses fisher information signals to trigger branching, replacing entropy spikes. The method is simple to plug into common RL algorithm.
- **Results on long-horizon benchmarks.** We evaluate GUA-RL on 13 challenging tasks in math, multi-hop QA, and deep search, and show consistent improvements over strong baselines.

II. Preliminaries

A. Agentic Reinforcement Learning with Tools

We consider an agentic reinforcement learning setting where a policy LLM interacts with an external tool environment over multiple turns. Each rollout starts from a task input x sample from dataset $\mathcal{D}$. The policy π_θ produces a sequence that interleaves: (i) natural-language reasoning , (ii) structured tool invocation actions, and (iii) the final answer. When a tool is called, the environment executes the corresponding API and returns an observation (tool feedback) that is appended to the context and conditions subsequent generation. This interaction structure matches standard *tool-augmented rollouts* used in recent agentic RL methods [10], [11].

Following prior agentic RL formulations [6], we optimize a KL-regularized objective:

$$\begin{aligned}\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x;\mathcal{T})}\big[r_\phi(x,y)\big] \\ -\beta\,\mathbb{D}_{\mathrm{KL}}(\pi_\theta(y\mid x;\mathcal{T})\,\|\,\pi_{\mathrm{ref}}(y\mid x;\mathcal{T}))\end{aligned} \tag{1}$$

where $\mathcal{T}$ denotes the set of available tools, r_ϕ is a verifiable reward, and π_{ref} is a fixed reference model used for stabilization.

A rollout can be decomposed into (a) an *agentic reasoning* segment that includes tool calls and tool feedback insertions, and (b) an *answer generation* segment conditioned on the resulting trajectory context. One convenient factorization is:

$$\begin{aligned}P_\theta(\mathcal{R}, y\mid x;\mathcal{T}) = \prod_{t=1}^{T_{\mathcal{R}}} P_\theta(\mathcal{R}_t\mid \mathcal{R}_{<t}, x;\mathcal{T}) \\ \cdot\prod_{t=1}^{T_y} P_\theta(y_t\mid y_{<t}, \mathcal{R}, x;\mathcal{T})\end{aligned} \tag{2}$$

where $\mathcal{R}$ denotes the tool-interleaved reasoning trajectory and y is the final answer.

B. Uncertainty Signals at Post-Tool States

Prior work [8]–[10] reports elevated token-level uncertainty immediately after tool interactions (Figure 1, b). Motivated by this observation, algorithm like ARPO uses entropy-based triggers to allocate additional branch rollouts in such rounds.

Formally, let $p_t = \pi_\theta(\cdot \mid \mathcal{R}_{<t}, x;\mathcal{T})$ be the next-token distribution at generation step t. The token entropy is:

$$H(p_t) = -\sum_{v=1}^{V} p_{t,v}\log p_{t,v} \tag{3}$$

While entropy is intuitive and cheap, it is a scalar statistic: different distributional regimes can yield similar entropy values. This motivates uncertainty measures that retain more information about *distribution geometry*.

To capture the geometric structure, we introduce the fisher information matrix associated with p_t. In sections III, we

derive efficient scalar summaries of $F(p_t)$ and show how they provide a efficient trigger for adaptive branching after tool interactions.

III. METHOD: GUA-RL

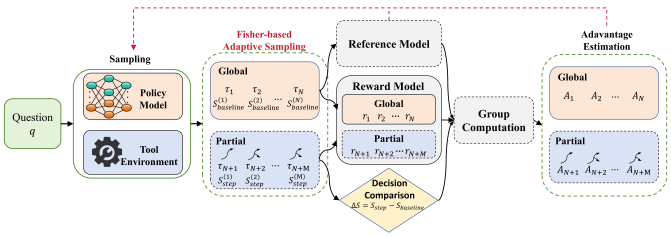


Fig. 2. The overview of GUA-RL algorithm

In this section, we introduce GUA-RL, which leverages the fisher information metric to capture the intrinsic geometric structure of the model's predictive distribution manifold. As illustrated in Figure 2, our method employs information geometric distance measurements to identify critical decision points for branch sampling.

Our rollout mechanism is motivated by the observation that external tool feedback can fundamentally reshape the geometry of the policy's predictive distribution. Entropy-based triggers summarize such changes with a single scalar, but they often miss how the distribution's shape changes after the feedback. To better capture these shifts, we measure changes in the predictive distribution using the fisher information metric. The procedure consists of three phases.

a) Global Initialization: Given a task input q comprising the user query and system instructions and global rollout size of M, the policy model π_θ performs N independent global samplings to generate initial trajectories $(\tau_1, \tau_2, \ldots, \tau_N)$. Each trajectory τ_i comprises reasoning text, tool invocation commands, and environmental feedback until termination or maximum length. During this global sampling, we establish a distributional baseline for each trajectory by computing fisher scores over an initial window of length W following the reception of q. Specifically, for trajectory i, we define its information geometric baseline as

$$S^{(i)}_{\text{baseline}} = \frac{1}{W}\sum_{t=1}^{W} S^{(i)}_t \tag{4}$$

where $S^{(i)}_t$ denotes the normalized fisher score at position t within trajectory i. This baseline captures the uncertainty geometry exhibited by the model when processing the input, thereby providing a reference for detecting relative geometric deviations induced by subsequent tool interactions.

b) Monitoring of Information Geometric Dynamics: After initialization, the model interacts with tools over multiple turns. GUA-RL will track how the policy's output distribution changes, and focus on the generations after tool feedback. Specifically, after the m-th tool call returns an observation o_m, GUA-RL start a detection window of K steps. Within this window, GUA-RL compute fisher scores online to measure how strongly the predictive distribution shifts.

At each position t in the window, GUA-RL obtain the next-token distribution $p_t = \text{softmax}(\mathbf{z}_t) \in \mathbb{R}^k$, where $\mathbf{z}_t$ represents the model's logit vector at position t and k denotes vocabulary size. We then compute the second and third moments:

$$s_t = \sum_{i=1}^{k} p_{t,i}^2 \quad t_t = \sum_{i=1}^{k} p_{t,i}^3 \tag{5}$$

which requires a single pass over the vocabulary ($\mathcal{O}(k)$). Using these moments, the frobenius norm of the fisher information matrix $\mathbf{F}_{p_t} = \text{diag}(p_t) - p_t p_t^\top$ can be computed without explicit matrix construction:

$$||\mathbf{F}_{p_t}||_F = \sqrt{\text{tr}(\mathbf{F}_{p_t}^T \mathbf{F}_{p_t})} = \sqrt{s_t + s_t^2 - 2t_t} \tag{6}$$

This avoids the $\mathcal{O}(k^2)$ cost and, once (s_t, u_t) are computed, reduces the remaining computation to $\mathcal{O}(1)$. To make scores comparable across different vocabulary sizes, we normalize by the fisher frobenius norm of the uniform distribution $\|\mathbf{F}_{\text{uni}}\|_F = \frac{\sqrt{k-1}}{k}$. For numerical stability, we use $k' = k + 1$ and define the normalized score:

$$S_t = \frac{||\mathbf{F}_{p_t}||_F}{\sqrt{(k'-1)/k'}} \in [0, 1] \tag{7}$$

Geometrically, larger S_t indicates a distribution closer to uniform, which means higher uncertainty, whereas smaller S_t corresponds to a more concentrated distribution, which is closer to deterministic.

To turn the K fisher scores in the detection window, $S_{t_0+1}, S_{t_0+2}, \ldots, S_{t_0+K}$ (where t_0 is the first position after tool feedback), into a step-level uncertainty score, we aggregate them:$S_{\text{step}} = \text{Agg}(S_{t_0+1}, \ldots, S_{t_0+K})$, where $\text{Agg}(\cdot)$ denotes a fixed aggregation operator.

c) Adaptive Branching via Geometric Deviation: Based on step-level score S_{step} computed in the detection window, we decide whether to branch at the current step.

$$\Delta S = S_{\text{step}} - S^{(i)}_{baseline} \tag{8}$$

When $\Delta S > 0$, the predictive distribution changes more strongly than the baseline, suggesting increased uncertainty after recent interactions (e.g., tool feedback). When $\Delta S < 0$, the distribution is closer to the baseline, indicating a more stable and confident regime.

We convert this deviation into an adaptive branching decision through a probability threshold mechanism:

$$\text{Action}_t = \begin{cases} \texttt{branch}, & \text{if } P_t = \alpha + \beta \cdot \Delta S > \gamma, \\ \texttt{continue}, & \text{otherwise.} \end{cases} \tag{9}$$

where α represents a base branching score and β controls sensitivity to geometric deviations. When P_t exceeds a predetermined threshold γ, the policy initiates Z additional partial sampling branches from the current node; otherwise,

sampling continues along the existing trajectory. This mechanism dynamically allocates exploration resources to regions where information geometric analysis reveals heightened distributional uncertainty, thereby efficiently discovering diverse tool-use behaviors.

The rollout stops branching when either: (1) the number of created branch paths $\hat{Z}$ reaches the partial-branch budget $M - N$, after which we finish the remaining rollout without further branching; or (2) all paths terminate early. In the second case, we add $M - N - \hat{Z}$ extra full-trajectory samples to meet the total budget.

Compared with entropy-based triggers, GUA-RL is driven by fisher geometric deviation, which reflects local changes of the predictive distribution beyond pure probability spread. This helps identify tool call steps where additional branching is more likely to improve exploration and learning signals. Algorithm 1 summarizes the procedure of GUA-RL.

Algorithm 1 GUA-RL: Information Geometric Uncertainty Modeling for Agentic Reinforcement Learning

Input: initial policy $\pi_{\theta_{\text{init}}}$, reward model r_ϕ, dataset $\mathcal{D}$, tools $\mathcal{T}$, hyperparameters $M, N, Z, W, K, \alpha, \beta, \gamma$ (total rollout budget, initial global samples, branches per trigger, baseline window, post-tool detection window, base branching probability, deviation sensitivity, threshold).
Output: optimized policy π_θ.

Initialize: $\pi_\theta \leftarrow \pi_{\theta_{\text{init}}}$
for each training iteration **do**
  $\pi_{\text{ref}} \leftarrow \pi_\theta$
  Sample a mini-batch of inputs $\mathcal{D}_b \subset \mathcal{D}$
  for each input $q \in \mathcal{D}_b$ **do**
    Sample N initial trajectories $\{\tau_i\}_{i=1}^{N} \sim \pi_\theta(\cdot \mid q; \mathcal{T})$
    Compute baseline score for each trajectory: $S_{\text{baseline}}^{(i)} \leftarrow \frac{1}{W}\sum_{t=1}^{W} S_t^{(i)}$
    $\mathcal{R} \leftarrow \{\tau_i\}_{i=1}^{N}$, $\hat{Z} \leftarrow 0$
    while there exists an unfinished trajectory in $\mathcal{R}$ **do**
      Execute next step; if a tool is called, receive observation $o_m = \mathcal{T}(\cdot)$ and append it to the context
      After each tool feedback, open a detection window of length K and compute the aggregated step score $S_{\text{step}} \leftarrow \text{Agg}(\{S_{t_0+1}, \ldots, S_{t_0+K}\})$
      $\Delta S \leftarrow S_{\text{step}} - S_{\text{baseline}}^{(i)}$
      $P \leftarrow \alpha + \beta \cdot \Delta S$
      if $P > \gamma$ **and** $\hat{Z} + Z \leq M - N$ **then**
        Branch Z additional partial rollouts from the current state and add them to $\mathcal{R}$
        $\hat{Z} \leftarrow \hat{Z} + Z$
      end if
      Continue rollout until termination or maximum length
    end while
    If $|\mathcal{R}| < M$, add $M - |\mathcal{R}|$ extra full-trajectory samples to meet budget
    Compute rewards $\{r_i\}$ for trajectories in $\mathcal{R}$ using r_ϕ
    Update π_θ with a trajectory-level RL optimizer (e.g., GRPO) under π_{ref}
  end for
end for
return π_θ

IV. Experiment Settings

A. Datasets

To comprehensively assess the effectiveness of our information geometric approach for training LLM-based tool-using agents, we evaluate it on three categories of long-horizon reasoning tasks, encompassing 13 challenging benchmarks.

Mathematical Reasoning. We evaluate computational reasoning capabilities on five datasets spanning different difficulty levels: AIME2024 and AIME2025, representing competition-level mathematical challenges; MATH500 [12], a curated subset of high-difficulty problems; the complete MATH dataset covering diverse mathematical domains [13]; and GSM8K [14], assessing fundamental arithmetic reasoning.

Knowledge-Intensive Reasoning. This category comprises five datasets requiring multi-hop reasoning over external knowledge sources: WebWalker [15], which demands strategic web navigation; and four Wikipedia-based open-domain question answering benchmarks—HotpotQA, 2WikiMultihopQA [16], Musique [17], and Bamboogle [18] — each requiring the integration of information across multiple retrieved documents.

Deep Search. For evaluating complex information seeking capabilities, we adopt three challenging benchmarks: General AI Assistant (GAIA) [19], which tests real-world assistant capabilities requiring extensive tool orchestration; WebWalker [15], evaluating strategic web exploration; and Humanity's Last Exam (HLE) [20], featuring exceptionally difficult questions designed to challenge frontier AI systems.

To ensure consistency with prior work in agentic reinforcement learning, we adopt the dataset splits established by Tool-Star for mathematical and knowledge-intensive reasoning benchmarks [21]. For deep search tasks, we follow the evaluation protocols defined by WebThinker [22] and HIRA [23].

B. Baselines

We compare our GUA-RL against three categories of baseline methods:

Direct Reasoning Models. For mathematical and knowledge-intensive reasoning, we evaluate instruction versions of Qwen2.5 [24]. For deep search tasks evaluated on Qwen3-8B [25], we additionally reference several advanced reasoning models: QwQ [26], DeepSeek-R1 [27], GPT-4o [28], and o1-preview [28], providing context for state-of-the-art performance levels.

Trajectory-level RL Algorithms. We benchmark against mainstream reinforcement learning algorithms designed for training LLM-based agents: GRPO [6], a group-relative policy optimization method; DAPO [29], which employs adaptive clipping strategies; and ARPO [8], the state-of-the-art entropy-based agentic RL algorithm that serves as our primary comparison baseline.

LLM-based Search Agents. For deep search benchmarks, we include workflow-based search agent implementations as additional references: vanilla RAG [30], representing standard retrieval-augmented generation; Search o1 [31], employing

TABLE I
OVERALL PERFORMANCE ON 10 CHALLENGING REASONING TASKS ARE PRESENTED. THE TOP TWO OUTCOMES ARE **BOLDED** AND UNDERLINED.

	Mathematical Reasoning					Knowledge-Intensive Reasoning					
Method	*AIME24*	*AIME25*	*MATH500*	*GSM8K*	*MATH*	*WebWalker*	*HotpotQA*	*2Wiki.*	*MuSiQue*	*Bamboogle*	**Avg.**
Qwen2.5-3B	3.3	3.3	52.8	62.1	54.5	5.7	11.9	8.3	3.6	11.2	21.7
+ TIR Prompting	6.7	6.7	42.8	47.1	41.8	15.0	16.7	14.4	6.2	16.6	21.4
+ GRPO	20.0	13.3	**72.0**	85.0	71.0	19.5	56.4	64.7	25.1	65.7	49.3
+ DAPO	20.0	16.7	71.4	84.0	**72.0**	19.5	54.0	61.5	**30.0**	66.8	49.6
+ ARPO	**23.3**	16.7	68.8	84.0	70.0	20.5	**58.5**	66.8	28.7	67.9	50.5
+ **GUA-RL**	**23.3**	**20.0**	70.4	**86.0**	**72.0**	**23.5**	**58.5**	**67.0**	28.5	**70.1**	**51.9**
Qwen2.5-7B	10.0	3.3	70.4	89.5	70.5	4.8	15.9	18.2	8.3	30.8	32.2
+ TIR Prompting	6.6	13.3	70.4	83.5	71.5	10.6	17.6	23.2	8.7	24.9	33.0
+ GRPO	26.7	23.3	78.0	90.8	78.8	19.5	59.0	77.8	29.8	68.4	55.2
+ DAPO	23.3	20.0	**80.4**	90.8	81.0	20.0	57.1	68.8	28.3	65.5	53.5
+ ARPO	30.0	**30.0**	78.2	**91.3**	81.0	19.4	**60.9**	76.5	29.9	67.9	56.5
+ **GUA-RL**	**33.3**	**30.0**	**80.4**	90.8	**81.5**	**20.7**	59.3	**78.5**	**30.5**	**73.6**	**57.9**

chain-of-thought search strategies; WebThinker [22], utilizing reflective reasoning over web content; and ReAct [32], combining reasoning traces with action execution.

C. Training

We use an open-access training pipeline based on VeRL [33] with a two-stage recipe: (i) cold-start via LLaMAFactory [34] to avoid early reward collapse, trained on 54k tool-integrated Tool-Star examples plus 800 STILL samples following CORT's curriculum design [35]; (ii) RL for tool-use exploration on 10k Tool-Star instances for math/knowledge tasks [21], and 1k deep-search instances from SimpleDeepSearcher [36] and WebSailor [37]. The environment provides top-10 Google Search snippets, a sandboxed Python interpreter, and token-level F1 rewards where applicable.

D. Evaluation Metric

During evaluation, we equip models with full search engine and web browsing capabilities to assess realistic reasoning performance. For accuracy measurement, we employ task-appropriate metrics: token-level F1 scores evaluate the four knowledge-intensive QA tasks, while all other tasks utilize LLM-as-Judge evaluation with Qwen2.5-72B-Instruct. Following prior work [31], we extract final answers enclosed in `\box{}` delimiters from model outputs. All experiments report pass@1 accuracy under with temperature set to 0.6 and nucleus sampling (top-p) set to 0.95.

V. EXPERIMENT RESULT

A. Performance on Various Benchmarks

To comprehensively assess the effectiveness of our information geometric approach, we evaluate GUA-RL against strong baselines across 13 challenging benchmarks spanning mathematical reasoning, knowledge-intensive reasoning, and deep search domains.

The main results on mathematical and knowledge-intensive reasoning tasks are presented in Table I. Across both benchmark categories and model backbones, GUA-RL demonstrates consistent improvements over trajectory-level RL algorithms, validating the effectiveness of fisher geometric adaptive exploration. We derive the following key insights:

GUA-RL Establishes State-of-the-art Performance. On the Qwen2.5-7B backbone, GUA-RL achieves the highest average performance across all 10 benchmarks, surpassing the previous best agentic RL method ARPO by 1.4%. Notably, GUA-RL delivers particularly strong gains on competition-level mathematical reasoning tasks and complex multi-hop QA benchmarks. This pattern suggests that fisher information geometry provides more precise uncertainty quantification than entropy-based methods, especially in scenarios requiring intricate tool orchestration and multi-step verification.

Fisher-based branching improves rollouts. Compared with ARPO's entropy trigger, GUA-RL uses fisher-based geometric signals to decide branching after tool feedback. Empirically, the largest gains appear on knowledge-intensive benchmarks such as Bamboogle (+5.7%) and 2WikiMultihopQA (+2.0%), where tool outputs (search/retrieval) often change what information is relevant for the next step. A plausible explanation is that fisher signals are more selective about when extra sampling is useful: they tend to trigger branching when tool feedback meaningfully changes the model's next-step preferences, and avoid oversampling when the model is simply diffuse. As a result, the same rollout budget is used more effectively, leading to stronger multi-hop reasoning and better use of retrieved evidence.

Consistent Gains Across Model Architectures. Qwen2.5-3B experiment show similar trends with Qwen2.5-7B, confirming that fisher geometric guidance provides robust benefits independent of model scale. This architectural consistency underscores the fundamental nature of information geometry in characterizing policy uncertainty during tool interactions.

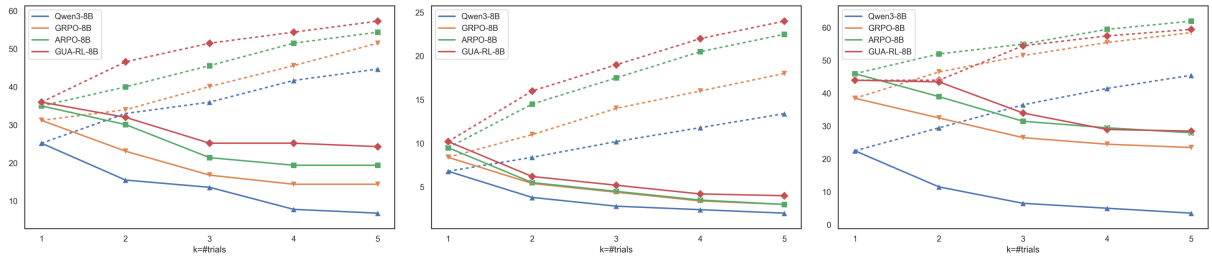
Table II presents performance on three challenging deep search benchmarks that demand extensive web exploration and multi-turn tool orchestration. GUA-RL demonstrates competitive performance against ARPO and significantly outperforms workflow-based agents and advanced reasoning models.

Importance of Step-Level Tool-Use Exploration. GUA-

TABLE II

OVERALL PERFORMANCE ON VARIOUS DEEP SEARCH TASKS, WITH ACCURACY RESULTS FOR EACH DATASET OBTAINED USING LLM-AS-JUDGE. THE BEST RESULTS ARE INDICATED IN **BOLD**, AND THE SECOND-BEST RESULTS ARE UNDERLINED.

Method	General AI Assistant				WebWalkerQA				Humanity's Last Exam				Avg.
	Lv.1	*Lv.2*	*Lv.3*	*Avg.*	*Easy*	*Med*	*Hard*	*Avg.*	*NS*	*CE*	*SF*	*Avg.*	
Direct Reasoning(≥32B)													
Qwen3-32B-thinking	26.2	12.1	0.0	14.9	6.9	1.1	2.9	3.1	14.6	9.8	8.4	12.6	10.2
DeepSeek-R1-32B	21.5	13.6	0.0	14.2	7.5	1.4	4.2	3.8	6.6	5.1	6.5	6.4	8.1
QwQ-32B	30.9	6.5	5.2	18.9	7.5	2.1	4.6	4.3	11.5	7.3	5.2	9.6	10.9
GPT-4o	23.1	15.4	8.3	17.5	6.7	6.0	4.2	5.5	2.7	1.2	3.2	2.6	8.5
DeepSeek-R1-671B	40.5	21.2	5.2	25.2	5.0	11.8	11.3	10.0	8.5	8.1	9.3	8.6	14.6
o1-preview	-	-	-	-	11.9	10.4	7.9	9.9	12.9	8.1	6.6	11.1	-
Single-Enhanced Method(Qwen3-8B)													
Vanilla RAG	28.2	15.4	**16.7**	20.4	8.9	10.7	9.9	10.0	5.1	1.6	12.9	5.8	12.1
Search-o1	35.9	15.4	0.0	21.4	6.7	15.5	9.7	11.5	7.6	2.7	5.3	6.4	13.1
WebThiker	43.6	11.5	0.0	22.3	6.7	13.1	16.9	13.0	7.3	4.0	6.3	6.6	14.0
ReAct	35.9	17.3	8.3	23.3	8.9	16.7	18.3	15.5	4.2	4.0	6.3	4.6	14.5
RL-based Method													
Qwen3-8B	30.8	26.9	0.0	25.2	17.8	28.2	22.6	23.5	5.2	6.9	8.0	6.8	18.5
+ GRPO	48.7	25.0	8.3	32.0	46.7	38.1	33.8	38.5	7.8	4.0	10.5	8.4	26.3
+ ARPO	53.9	**40.4**	0.0	40.8	44.4	**40.9**	**46.4**	**44.0**	7.2	9.0	13.7	9.5	31.4
+ **GUA-RL**	**61.5**	34.6	8.3	**41.7**	**48.9**	39.4	44.0	43.5	**7.9**	**9.7**	**14.0**	**10.2**	**31.8**



(a) GAIA (b) HLE (c) WebwalkerQA

Fig. 3. Performance across trials: passˆk (-) and pass@k (...)

RL delivers consistent improvements over other RL baselines and remains competitive with strong agentic RL methods across three challenging deep-search tasks. The gains are especially clear on GAIA and WebWalkerQA: GUA-RL raises GAIA average accuracy from 32.0% (GRPO) to 41.7% (+9.7%), and improves WebWalkerQA from 38.5% to 43.5% (+5.0%). Compared with GRPO, GUA-RL uses step-level branching after tool feedback to focus samples on the decisions that matter most, improving overall accuracy and making performance more stable on hard information seeking tasks.

Generalization in Deep Search with Efficient Exploration. Deep search benchmarks remain challenging even for frontier-scale models, which achieve only modest accuracy across these tasks. In contrast, GUA-RL trained on Qwen3-8B attains stronger and more consistent performance, matching or surpassing much larger models in overall results. This comparison highlights an important takeaway: success in web-scale, multi-turn information seeking depends not only on model capacity, but also on how effectively the agent explores and updates its behavior after tool interactions. By using fisher geometric signals to allocate branching and sampling to the most influential steps, GUA-RL improves long-horizon tool-use behaviors and helps smaller open models reach competitive performance without relying on massive model scaling.

B. Quantitative Analysis

Sampling Analysis. Figure 3 reports multi-trial performance on three representative agentic benchmarks (GAIA, HLE, and WebWalkerQA), where dashed curves denote pass@k and solid curves denote passˆk [38]. Here, passˆk reflects execution consistency: it measures the probability that

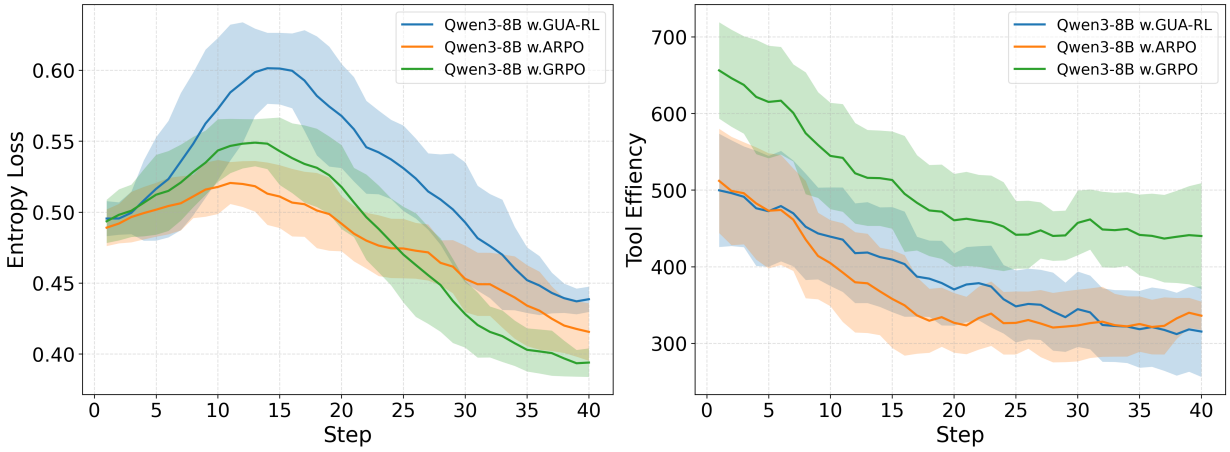


Fig. 4. Visualization of training dynamics, including entropy loss(left) and comparison of tool call efficiency(right)

an agent can solve the same task successfully in all of k independent runs, thus capturing robustness under conversational and tool-interaction variations rather than merely the average success in a single attempt. In all three settings, we observe a clear divergence between pass@k and passˆk as k increases, indicating that improvements in "at least one success within k trials" do not necessarily translate into "reliably succeeding every time" for the same task.

Across GAIA and HLE, GUA-RL exhibits a clear advantage in sampling scalability, showing stronger growth of Pass@k as k increases, while also attenuating the typical degradation of passˆk compared to other baselines, indicating improved consistency under conversational variations. On WebWalkerQA, although ARPO attains the best Pass@k at larger k, GUA-RL remains competitive and achieves comparable or slightly stronger—reliability at higher k, suggesting that its gains are more pronounced in stabilizing repeated successes rather than uniformly maximizing best-of-k outcomes.

Entropy Stability Analysis. To validate the effectiveness of GUA-RL during policy optimization, we visualize the search task training dynamics of entropy loss in Figure 4 (left). The curve reveals a structured evolution pattern: entropy loss increases in the early stage, indicating that the policy retains sufficient stochasticity to continue exploring alternative reasoning and tool-use decisions, and then gradually decreases as learning progresses, reflecting a controlled transition toward more confident and consistent action selection. Compared with GRPO and ARPO, GUA-RL exhibits a more pronounced mid training elevation of entropy loss and maintains a smoother downward trend thereafter, suggesting that GUA-RL induces a stable uncertainty profile during updates instead of relying on transient entropy spikes. This pattern matches the purpose of GUA-RL: it promotes exploration when uncertainty is useful and avoids unstable updates. Overall, it suggests that GUA-RL supports steady learning in multi-turn decision making.

Tool call Efficiency Analysis. Since tool usage is a major practical constraint for agent training and deployment, Figure 4 (right) further reports the evolution of tool efficiency throughout training. To evaluate tool call efficiency, we compare GUA-RL with GRPO and ARPO under the same backbone setting. As shown in Figure 4 (right), GUA-RL reaches higher overall task performance while requiring fewer tool interactions over the course of training, and its tool call count drops steadily before stabilizing. This indicates that GUA-RL learns to solve tasks with less reliance on tools, rather than expanding rollouts through frequent tool invocations. This efficiency comes from GUA-RL's geometry-aware branching rule. Instead of sampling more whenever the policy is broadly uncertain, GUA-RL allocates extra branches mainly at tool-feedback steps where the predictive distribution shifts most, which are more likely to change downstream decisions. As a result, GUA-RL expands exploration of tool-use behaviors at the right moments, while reducing redundant or low-gain tool calls compared with GRPO.

VI. Conclusion

In this paper, we present GUA-RL, an agentic reinforcement learning framework tailored for tool-use scenarios, addressing a key limitation of prior entropy-guided methods: while token entropy effectively highlights high-uncertainty phases after tool interactions, as a scalar measure of dispersion it fails to capture the geometric structure of predictive distributions and cannot reliably distinguish informed uncertainty from uninformative flatness. To address this, GUA-RL replaces entropy-based branching with fisher information geometry, using the fisher information matrix to catch uncertainty; by monitoring fisher geometric dynamics within post-tool detection windows, it triggers adaptive branching and sampling only when tool feedback induces meaningful geometric deviations with high exploration potential. Experiments across 13 challenging benchmarks spanning computational reasoning, knowledge reasoning, and deep search show that GUA-RL consistently outperforms strong baselines. It also offers clear advantages in sampling scalability and tool-use efficiency making it a scalable option for aligning LLM-based agents in dynamic environments.

VII. Acknowledgement

This work was supported by the International Partnership Program of the Chinese Academy of Sciences (Grant No.159231KYSB20200010)

References

[1] J. Feng, S. Huang, X. Qu, G. Zhang, Y. Qin, B. Zhong, C. Jiang, J. Chi, and W. Zhong, "Retool: Reinforcement learning for strategic tool use in llms," *arXiv preprint arXiv:2504.11536*, 2025.

[2] D. Jiang, Y. Lu, Z. Li, Z. Lyu, P. Nie, H. Wang, A. Su, H. Chen, K. Zou, C. Du *et al.*, "Verltool: Towards holistic agentic reinforcement learning with tool use," *arXiv preprint arXiv:2509.01055*, 2025.

[3] J. Luo, M. Cheng, F. Wan, N. Li, X. Xia, S. Tian, T. Bian, H. Wang, H. Fu, and Y. Tao, "Globalrag: Enhancing global reasoning in multi-hop question answering via reinforcement learning," *arXiv preprint arXiv:2510.20548*, 2025.

[4] Z. Wei, W. Yao, Y. Liu, W. Zhang, Q. Lu, L. Qiu, C. Yu, P. Xu, C. Zhang, B. Yin *et al.*, "Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning," *arXiv preprint arXiv:2505.16421*, 2025.

[5] Z. Qi, X. Liu, I. L. Iong, H. Lai, X. Sun, J. Sun, X. Yang, Y. Yang, S. Yao, W. Xu *et al.*, "Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning," in *ICLR*, 2025.

[6] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu *et al.*, "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," *arXiv preprint arXiv:2402.03300*, 2024.

[7] L. Feng, Z. Xue, T. Liu, and B. An, "Group-in-group policy optimization for llm agent training," *arXiv preprint arXiv:2505.10978*, 2025.

[8] G. Dong, H. Mao, K. Ma, L. Bao, Y. Chen, Z. Wang, Z. Chen, J. Du, H. Wang, F. Zhang *et al.*, "Agentic reinforced policy optimization," *arXiv preprint arXiv:2507.19849*, 2025.

[9] M. Chen, G. Chen, W. Wang, and Y. Yang, "Seed-grpo: Semantic entropy enhanced grpo for uncertainty-aware policy optimization," *arXiv preprint arXiv:2505.12346*, 2025.

[10] G. Dong, L. Bao, Z. Wang, K. Zhao, X. Li, J. Jin, J. Yang, H. Mao, F. Zhang, K. Gai *et al.*, "Agentic entropy-balanced policy optimization," *arXiv preprint arXiv:2510.14545*, 2025.

[11] W. Zhao, X. Wang, C. Ma, L. Kong, Z. Yang, M. Tuo, X. Shi, Y. Zhai, and X. Cai, "Mua-rl: Multi-turn user-interacting agent reinforcement learning for agentic tool use," *arXiv preprint arXiv:2508.18669*, 2025.

[12] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe, "Let's verify step by step," in *The Twelfth International Conference on Learning Representations*, 2024.

[13] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt, "Measuring mathematical problem solving with the math dataset," *arXiv preprint arXiv:2103.03874*, 2021.

[14] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano *et al.*, "Training verifiers to solve math word problems," *arXiv preprint arXiv:2110.14168*, 2021.

[15] J. Wu, W. Yin, Y. Jiang, Z. Wang, Z. Xi, R. Fang, L. Zhang, Y. He, D. Zhou, P. Xie *et al.*, "Webwalker: Benchmarking llms in web traversal," *arXiv preprint arXiv:2501.07572*, 2025.

[16] X. Ho, A.-K. D. Nguyen, S. Sugawara, and A. Aizawa, "Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps," *arXiv preprint arXiv:2011.01060*, 2020.

[17] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "Musique: Multihop questions via single-hop question composition," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 539–554, 2022.

[18] O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, and M. Lewis, "Measuring and narrowing the compositionality gap in language models," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 5687–5711.

[19] G. Mialon, C. Fourrier, T. Wolf, Y. LeCun, and T. Scialom, "Gaia: a benchmark for general ai assistants," in *The Twelfth International Conference on Learning Representations*, 2024.

[20] L. Phan, A. Gatti, Z. Han, N. Li, J. Hu, H. Zhang, C. B. C. Zhang, M. Shaaban, J. Ling, S. Shi *et al.*, "Humanity's last exam," *arXiv preprint arXiv:2501.14249*, 2025.

[21] G. Dong, Y. Chen, X. Li, J. Jin, H. Qian, Y. Zhu, H. Mao, G. Zhou, Z. Dou, and J.-R. Wen, "Tool-star: Empowering llm-brained multi-tool reasoner via reinforcement learning," *arXiv preprint arXiv:2505.16410*, 2025.

[22] X. Li, J. Jin, G. Dong, H. Qian, Y. Wu, J.-R. Wen, Y. Zhu, and Z. Dou, "Webthinker: Empowering large reasoning models with deep research capability," *arXiv preprint arXiv:2504.21776*, 2025.

[23] J. Jin, X. Li, G. Dong, Y. Zhang, Y. Zhu, Y. Zhao, H. Qian, and Z. Dou, "Decoupled planning and execution: A hierarchical reasoning framework for deep search," *arXiv preprint arXiv:2507.02652*, 2025.

[24] Qwen, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu, "Qwen2.5 Technical Report," Jan. 2025, arXiv:2412.15115 [cs]. [Online]. Available: http://arxiv.org/abs/2412.15115

[25] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv *et al.*, "Qwen3 technical report," *arXiv preprint arXiv:2505.09388*, 2025.

[26] Q. Team, "Qwq: Reflect deeply on the boundaries of the unknown," *Hugging Face*, 2024.

[27] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[28] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford *et al.*, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.

[29] Q. Yu, Z. Zhang, R. Zhu, Y. Yuan, X. Zuo, Y. Yue, W. Dai, T. Fan, G. Liu, L. Liu *et al.*, "Dapo: An open-source llm reinforcement learning system at scale," *arXiv preprint arXiv:2503.14476*, 2025.

[30] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[31] X. Li, G. Dong, J. Jin, Y. Zhang, Y. Zhou, Y. Zhu, P. Zhang, and Z. Dou, "Search-o1: Agentic search-enhanced large reasoning models," *arXiv preprint arXiv:2501.05366*, 2025.

[32] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *The eleventh international conference on learning representations*, 2022.

[33] G. Sheng, C. Zhang, Z. Ye, X. Wu, W. Zhang, R. Zhang, Y. Peng, H. Lin, and C. Wu, "Hybridflow: A flexible and efficient rlhf framework," *arXiv preprint arXiv: 2409.19256*, 2024.

[34] Y. Zheng, R. Zhang, J. Zhang, Y. Ye, Z. Luo, Z. Feng, and Y. Ma, "Llamafactory: Unified efficient fine-tuning of 100+ language models," *arXiv preprint arXiv:2403.13372*, 2024.

[35] C. Li, G. Dong, M. Xue, R. Peng, X. Wang, and D. Liu, "Dotamath: Decomposition of thought with code assistance and self-correction for mathematical reasoning," *arXiv preprint arXiv:2407.04078*, 2024.

[36] S. Sun, H. Song, Y. Wang, R. Ren, J. Jiang, J. Zhang, F. Bai, J. Deng, W. X. Zhao, Z. Liu *et al.*, "Simpledeepsearcher: Deep information seeking via web-powered reasoning trajectory synthesis," *arXiv preprint arXiv:2505.16834*, 2025.

[37] K. Li, Z. Zhang, H. Yin, L. Zhang, L. Ou, J. Wu, W. Yin, B. Li, Z. Tao, X. Wang *et al.*, "Websailor: Navigating super-human reasoning for web agent," *arXiv preprint arXiv:2507.02592*, 2025.

[38] S. Yao, N. Shinn, P. Razavi, and K. Narasimhan, "Tau-bench: A benchmark for tool-agent-user interaction in real-world domains," *arXiv preprint arXiv:2406.12045*, 2024.