ArenaRL: Scaling RL for Open-Ended Agents via Tournament-based Relative Ranking

Qiang Zhang[1,†], Boli Chen[1], Fanrui Zhang[1,†], Ruixue Ding[1*], Shihang Wang[1], Qiuchen Wang[1] Yinfeng Huang[2], Haonan Zhang[2], Rongxiang Zhu[2], Pengyong Wang[2], Ailin Ren[2], Xin Li[2] Pengjun Xie[1], Jiawei Liu[‡], Ning Guo[2], Jingren Zhou[1], Zheng-Jun Zha

[1] Tongyi Lab, Alibaba Group [2] Amap, Alibaba Group

arXiv:2601.06487v3 [cs.LG] 22 Jul 2026

Abstract

Reinforcement learning (RL) has substantially improved the performance of large language model (LLM) agents on tasks with verifiable outcomes, such as mathematics and code generation, but it still struggles on open-ended agent tasks with vast solution spaces (*e.g.*, complex travel planning). Due to the absence of objective ground-truth for these tasks, current RL algorithms largely rely on reward models that assign scalar scores to individual responses. We contend that such pointwise scoring suffers from an inherent discrimination collapse: the reward model struggles to distinguish subtle advantages among different trajectories, resulting in scores within a group being compressed into a narrow range. Consequently, the effective reward signal becomes dominated by noise from the reward model, leading to optimization stagnation. To tackle this issue, we propose ArenaRL, a reinforcement learning paradigm that shifts from pointwise scalar scoring to intra-group relative ranking. ArenaRL introduces a process-aware pairwise evaluation mechanism, employing multi-level rubrics to assign fine-grained relative scores to trajectories. Additionally, we construct an intra-group adversarial arena and devise a tournament-based ranking scheme to obtain stable advantage signals. Empirical results confirm that the built seeded single-elimination scheme achieves nearly equivalent advantage estimation accuracy to full pairwise comparisons with $\mathcal{O}(N^2)$ complexity, while operating with only $\mathcal{O}(N)$ complexity, striking an optimal balance between efficiency and precision. Furthermore, to address the lack of full-cycle benchmarks for open-ended agents, we build Open-Travel and Open-DeepResearch, two high-quality benchmarks featuring a comprehensive pipeline covering supervised fine-tuning, RL training, and multi-dimensional automated evaluation. Extensive experiments show that ArenaRL substantially outperforms standard RL baselines, enabling LLM agents to generate solutions that are more logically rigorous and robust on complex real-world tasks. The code is available at https://github.com/Alibaba-NLP/qqr.

1 Introduction

The evolution of large language models (LLMs) into autonomous agents marks a paradigm shift in artificial intelligence from passive question answering to active problem solving. By integrating long-horizon planning and tool use, such agents have demonstrated substantial potential in handling complex tasks Yao et al. (2022b); Li et al. (2025a); Team et al. (2025). In this progression, reinforcement learning (RL) has played a pivotal role, particularly in deterministic tasks such as mathematical reasoning and code generation, where ground-truth labels provide explicit reward signals for optimization Dong et al. (2025); Li et al. (2025b). However, extending RL to open-ended agent tasks in real-world scenarios, such as personalized travel planning or in-depth industry analysis, poses fundamental challenges Li et al. (2025c). In these domains, the solution spaces are vast and

[†] Equal contribution, [*] Project leader, [‡] Corresponding author

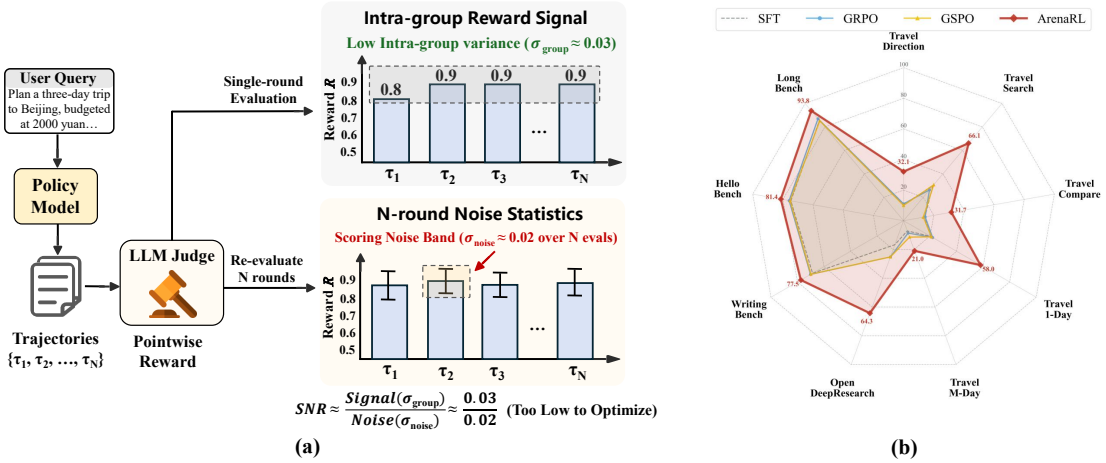

Figure 1: **(a) Illustration of discriminative collapse in pointwise evaluation:** We analyse the rewards of trajectory groups generated for a query example during RL training through two evaluation settings. First, the intra-group reward signal represents the pointwise rewards assigned to each trajectory within the group during a single-round evaluation. And the intra-group variance, denoted as σ_{group}, quantifies the degree of variation among different trajectories. Secondly, the N-round noise statistic present the average reward and corresponding scoring noise band for each trajectory across N independent evaluation repetitions, from which the noise variance σ_{noise} is estimated. Observations reveal that the evaluation noise variance σ_{noise} is substantial, comparable to the intra-group variance σ_{group}. This results in an extremely low signal-to-noise ratio (SNR), causing genuine advantages to be obscured by noise and hindering effective reinforcement learning optimization. **(b) ArenaRL Performance:** By shifting from pointwise scalar scoring to tournament-based relative ranking, ArenaRL significantly outperforms baselines (SFT, GRPO, GSPO) across diverse open-ended benchmarks.

unstructured, and the notion of correctness is inherently subjective and multi-dimensional, rendering traditional ground-truth based RL inapplicable Jia et al. (2025).

Mainstream RL approaches (*e.g.*, GRPO Shao et al. (2024), GSPO Zheng et al. (2025)) typically assume the existence of a verifiable reward function that can provide accurate reward feedback. However, this assumption often breaks down in open-ended settings where no objective ground truth is available. To address the issue of reward acquisition, recent work has adopted the LLM-as-Judge paradigm, assigning pointwise scalar scores to model outputs Viswanathan et al. (2025); Huang et al. (2025); Liu et al. (2025). We identify that this mechanism leads to a severe phenomenon that we term discriminative collapse, as shown in Figure 1(a). As the policy is progressively refined, the generated trajectories become increasingly similar in distribution. Consequently, rewards for trajectories within the same group are compressed by the pointwise scoring scheme into a narrow range (*e.g.*, 0.8–0.9 on a scale of 1), rendering rewards indistinguishable. Moreover, due to inherent noise in the LLM judge, such as decoding randomness Arias et al. (2025) and length preferences Hu et al. (2024), the reward outcomes exhibit a certain degree of unreliability, with a low signal-to-noise ratio between the reward signal and the interfering noise. In this situation, the pointwise evaluation mechanism struggles to distinguish truly superior samples. And the RL optimization process is driven more by spurious noise than by meaningful task-specific rewards, leading performance to stagnate or even deteriorate.

To fundamentally address discriminative collapse, we draw inspiration from decision theory, where pairwise preference judgments are known to be more stable than pointwise quantitative assessments Fürnkranz & Hüllermeier (2010); Rafailov et al. (2023), and advocate a paradigm shift from pointwise scalar scoring to intra-group relative ranking. To this end, we propose ArenaRL, an online policy optimization framework grounded in an adversarial arena. ArenaRL abandons unstable scalar

rewards in favor of constructing relative rankings over trajectories within each generated group. To ensure both depth and fairness in evaluation, we introduce a process-aware pairwise evaluation mechanism that not only compares the reliability of the outcome, but also scrutinizes the logical coherence of the chain-of-thought and the effectiveness of tool invocations along the trajectory.

A central bottleneck in scaling pairwise preference optimization to open-ended agent tasks lies in the high computational cost. While exhaustive comparisons yield accurate rankings, the resulting $\mathcal{O}(N^2)$ complexity is intractable for online training. To investigate the trade-off between ranking fidelity and sample efficiency, we designed and implemented five tournament topologies, ranging from exhaustive round-robin to single and double elimination formats. Our empirical analysis reveals a critical challenge: standard single and double tournaments are highly sensitive to the initial pairing combinations. Random matching causes premature encounters and eliminations of high-quality trajectories, degrading overall ranking accuracy. To mitigate this issue, we innovatively propose a seeded single-elimination mechanism. This approach utilizes the trajectory generated by greedy decoding as a "quality anchor" for pre-ranking, providing a low-biased initial estimate before seeding the tournament, after which a binary-tree structure is used for efficient ranking. Experiments show that this design reduces the computational complexity to linear $\mathcal{O}(N)$ while robustly preserving the accuracy of intra-group relative ranking, thereby achieving an optimal balance between training efficiency and advantage estimation fidelity.

Furthermore, to address the lack of full-cycle benchmarks for open-ended agents, we introduce two comprehensive benchmark suites: (1) Open-Travel, which focuses on evaluating agents' long-horizon planning capabilities under multiple hard constraints such as budget and spatiotemporal windows; and (2) Open-DeepResearch, which centers on assessing agents' abilities in autonomous information retrieval, and report generation in realistic internet environments. Unlike traditional benchmarks that only provide a test set He et al. (2025); Du et al. (2025); Coelho et al. (2025), the two proposed benchmarks offer a complete pipeline from supervised fine-tuning (SFT) $\rightarrow$ RL training $\rightarrow$ multi-dimensional automated evaluation, establishing a reproducible infrastructure for the community. Given the shared characteristics of open-ended tasks, we further extend our experiments to standard open-ended writing tasks Wang et al. (2025a) and conduct systematic evaluations on three public benchmarks. As demonstrated in Figure 1(c), ArenaRL yields substantial performance gains over strong baselines across travel planning, deep research, and open-ended writing tasks, validating the superiority of the tournament-based ranking paradigm.

In summary, our major contributions are as follows:

- We identify and formalize the problem of discriminative collapse in open-ended tasks, and propose ArenaRL, which replaces unstable pointwise scalar rewards with a tournament-based relative ranking mechanism to enable robust policy optimization.
- To address the high computational cost of pairwise comparisons, we design and validate a seeded single-elimination tournament topology that achieves high-accuracy advantage estimation with only $\mathcal{O}(N)$ complexity.
- We construct the Open-Travel and Open-DeepResearch benchmarks with full training pipelines, filling a critical gap in evaluating the full lifecycle of open-ended agents.

2 Related Work

Open-Ended Agent Benchmark. The rapid development of LLMs has given rise to autonomous agents that interact with external environments and solve complex tasks Guo et al. (2025); Yang et al. (2025). Most existing benchmark studies in this area focus on deterministic settings, where tasks have clear goals and verifiable feedback Phan et al. (2025); Yang et al. (2018). Benchmarks such as WebShop Yao et al. (2022a), Mind2Web Deng et al. (2023), and SWE-bench Jimenez et al. (2023) have advanced research in web navigation and code generation by testing whether agents can reach target pages or produce automatically verifiable code. In contrast, many critical real-world applications, such as personalized travel planning Ning et al. (2025) and in-depth industry

analysis Li et al. (2025c), are inherently open-ended and unstructured. These tasks rarely have a single gold solution; their quality depends on multi-dimensional trade-offs among the soundness of reasoning, satisfaction of personalized constraints, and practical usefulness of the final plan. More importantly, there is still a lack of a systematic training–evaluation infrastructure tailored to open-ended agentic tasks. Existing open-ended benchmarks (*e.g.*, VitaBench He et al. (2025), DeepResearchBench Du et al. (2025)) are predominantly static test suites that only support post-hoc evaluation. They lack complementary training pipelines to enable continuous improvement of open-ended agents. To address this gap, we introduce two high-quality benchmarks, Open-Travel and Open-DeepResearch. Unlike traditional benchmarks, they offer an integrated pipeline spanning SFT, RL-based exploration, and multi-dimensional evaluation, enabling systematic study of agent capabilities in creative, open-ended environments.

Reinforcement Learning with LLMs. Reinforcement learning (RL) has emerged as a powerful paradigm for aligning LLMs with complex objectives Zhang et al. (2025); Gao et al. (2025). In domains with clear ground truth (*e.g.*, math and code), rule-based reward let algorithms like GRPO Shao et al. (2024) and DAPO Yu et al. (2025) to achieve remarkable success Dong et al. (2025). However, extending RL to open-ended tasks that lack objective outcomes remains highly challenging both theoretically and practically. Because the notion of correctness in such tasks is inherently subjective and multi-dimensional, mainstream methods follow the LLM-as-Judge paradigm. They specify rubrics and assign scalar scores to individual trajectories Viswanathan et al. (2025); Huang et al. (2025) or enforce multiple constraints to assess reliability Ning et al. (2025). Yet, this reward mechanism often struggles to distinguish fine-grained differences among high-quality trajectories in open-ended settings. To address this, recent work has begun to explore comparison-based evaluation mechanisms. Writing-Zero Jia et al. (2025) assigns binary positive/negative advantages by comparing responses against random references, improving performance on open-ended writing tasks. Pref-GRPO Wang et al. (2025b) derives reward based on win rates via exhaustive pairwise comparison and shows promise on text to image tasks. Despite these advances, contrastive mechanisms remain unexplored for long-horizon agent tasks. Moreover, existing methods have inherent drawbacks: Writing-Zero provides only coarse-grained binary guidance, while the $\mathcal{O}(N^2)$ computational cost of full pairwise comparison is prohibitive for training long-context agents. Motivated by this, ArenaRL proposes a sparse tournament-based relative ranking paradigm. With an optimized tournament topology, ArenaRL maintains linear $\mathcal{O}(N)$ complexity while achieving high-accuracy advantage estimation.

3 Preliminary

In this section, we formally define the open-ended agentic task and the associated reinforcement learning objective. We then critically examine the limitations of pointwise scalar evaluation in this setting. Finally, we introduce a process-aware pairwise evaluation mechanism that enables fine-grained comparison between two trajectories and produces separate scores for each.

3.1 Task Definition

We formulate the open-ended agentic task as a conditional trajectory generation problem. Let T denote the set of accessible tools. Given a query x sampled from a task distribution $\mathcal{D}$, the agent policy π_θ synthesizes a multi-step interaction trajectory τ. Formally, τ is defined as an interleaved sequence comprising chain-of-thought z_k, tool invocations $a_k \in T$, environmental feedback o_k, and a final answer y:

$$\tau = [z_1, a_1, o_1, \ldots, z_k, a_k, o_k, \ldots, z_K, a_K, o_K, y] \tag{1}$$

The reinforcement learning objective is to align the agent's behavior with a reward signal while maintaining stability relative to a reference model. We formulate the optimization objective as follows:

$$\mathcal{L}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\tau\sim\pi_\theta(\cdot|x;T)}\Big[r_\phi(x,\tau) - \beta\mathbb{D}_{\mathrm{KL}}\big(\pi_\theta(\cdot|x) \parallel \pi_{\mathrm{ref}}(\cdot|x)\big)\Big] \tag{2}$$

where $r_\phi(x,\tau)$ represents the reward signal evaluating the quality of trajectory τ, π_{ref} is the reference policy, and β is a coefficient controlling the regularization strength of the KL divergence $\mathbb{D}_{\mathrm{KL}}$ to prevent policy degradation.

3.2 Pointwise Scalar evaluation

In open-ended agentic tasks, the ground-truth reward function R^* is intractable due to the absence of verifiable rules. Consequently, existing RL paradigms rely on an LLM as a reward model to assign pointwise scalar score to the final answer y of trajectory τ. We model the observed score $\hat{R}(\tau)$ as the true utility $R^*(\tau)$ corrupted by noise ϵ.

The reliance on such pointwise scalar feedback presents a critical vulnerability: discriminative collapse. As the agent's capabilities improve, its generated responses $\{\tau_i\}_{i=1}^G$ tend to converge within a narrow band of high-quality solutions, causing the variance within the group to vanish ($\sigma_{\mathrm{group}} \to 0$). In this regime, the reward model suffers from high epistemic uncertainty. Unable to discern subtle advantages between top-tier candidate trajectories, the judge hesitates, exhibiting high-variance scoring behavior driven by spurious correlations such as length preference or decoding stochasticity, rather than by genuine semantic merit.

This scoring criterion drift, caused by the judge's lack of discriminative ability, manifests macroscopically as high-amplitude noise ϵ. Standard algorithms like GRPO, which normalize scores based on group variance ($A_i = (\hat{R}_i - \mu)/\sigma$), catastrophically fail in this scenario. As σ_{group} vanishes, the normalization term inadvertently amplifies this drift-induced noise into dominant gradient signals. Consequently, the optimization process is hijacked by the reward model's interfering noise, leading to performance stagnation or even degeneration.

3.3 Process-Aware Pairwise Evaluation

To circumvent the pitfalls of pointwise scalar scoring, we pivot to a pairwise comparison–based evaluation paradigm. We construct an Arena Judge, denoted as $\mathcal{J}$. Given a query x and two candidate trajectories τ_a, τ_b, the judge $\mathcal{J}$ evaluates them jointly and outputs a separate quality score for each trajectory.

Specifically, the input to $\mathcal{J}$ consists of three components: (1) the user query x; (2) the core context of trajectories τ_i and τ_j (containing the chain-of-thought z_k and tool invocations a_k for each step, and the final answer y); and (3) a comprehensive process-aware rubric u. This rubric enforces fine-grained scrutiny of logical consistency in the chain-of-thought, the precision of tool calls, and the reliability of the final answer. This ensures that the optimization signal reinforces the agent's intrinsic reasoning capabilities rather than merely overfitting to surface-level features of the final answer. The detailed prompts are provided in the Appendix E.

To mitigate the positional bias in LLM judges Wu et al. (2025a), we employ a bidirectional scoring protocol. We conduct two independent evaluations by swapping the presentation order of the trajectories:

$$(s_i, s_j) = \mathcal{J}(x, \tau_i, \tau_j, u) + \mathcal{J}(x, \tau_j, \tau_i, u) \tag{3}$$

where s_i, s_j denotes the quality score assigned to τ_i in the bidirectional pairwise evaluation, which eliminates the bias favoring the first or second position.

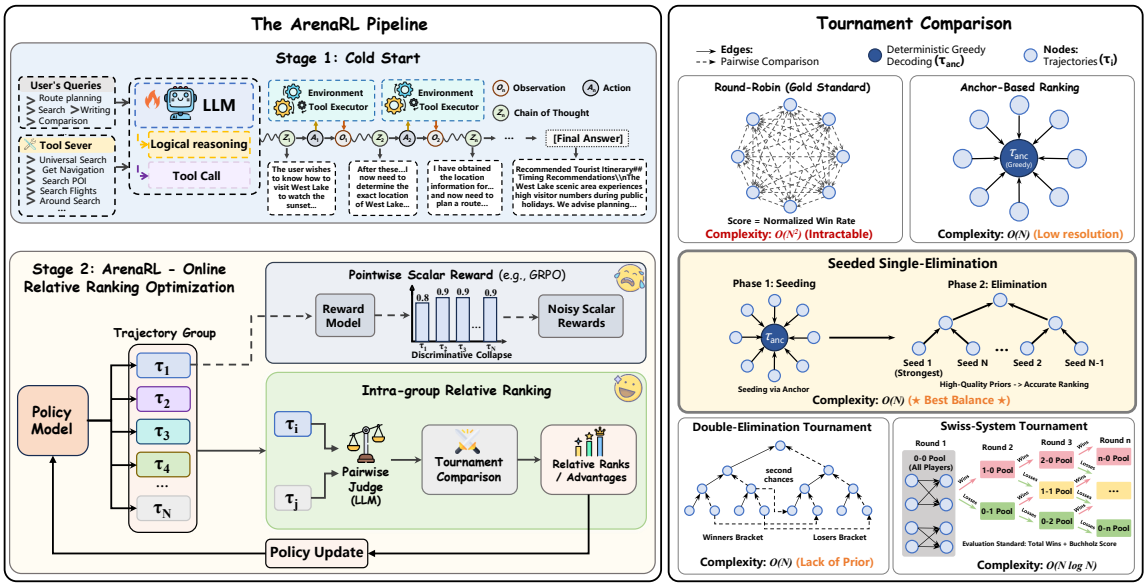

Figure 2: The overall of the proposed ArenaRL algorithm. ArenaRL replaces conventional pointwise scalar reward paradigm with intra-group relative ranking and designs five distinct tournament topologies to optimally balance training efficiency against the accuracy of advantage estimation.

4 Methodology: ArenaRL

In this section, we details the proposed ArenaRL algorithm, as shown in Figure 2. Departing from the reliance on unstable pointwise scalar rewards, ArenaRL redefines reward modelling as a relative quality ranking among trajectories within a group. The core tenet is to efficiently construct a dynamic arena for the trajectory group $\mathcal{G} = \{\tau_1, \tau_2, \ldots, \tau_N\}$ sampled from the current policy π_θ, thereby deriving robust advantage signals. We systematically investigate five tournament topologies to identify the optimal trade-off between computational cost and ranking fidelity.

4.1 Round-Robin Tournament

In this scheme, every trajectory τ_i competes against all other $N-1$ trajectories via our process-aware pairwise evaluation. The final score is defined as the normalized win rate:

$$\text{Score}(\tau_i) = \frac{1}{N-1} \sum_{j \neq i} \mathbb{I}(s_i > s_j) \tag{4}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function, which equals 1 if the score s_i (from the process-aware pairwise evaluation) is greater than s_j, and 0 otherwise. The group ranking is determined by sorting the $\text{Score}(\tau_i)$ in descending order. While the Round-Robin can theoretically provide unbiased intra-group rankings, its quadratic complexity $\mathcal{O}(N^2)$ renders it intractable for online training with a large group size N. We primarily utilize it as the "gold standard" to benchmark the fidelity of other efficient topologies.

4.2 Anchor-Based Ranking

To alleviate computational complexity, we introduce an Anchor-Based Ranking mechanism. For a given input x, we first generate a deterministic reference trajectory, denoted as the quality anchor τ_{anc}, using greedy decoding (Temperature=0). The remaining $N-1$ trajectories τ_i, $i = \{1, 2, ..., N-1\}$

in group $\mathcal{G}$ are produced via high-entropy sampling (*e.g.*, Temperature=0.8) to ensure exploration diversity.

Subsequently, each exploratory trajectory τ_i is individually compared with the anchor trajectory τ_{anc}, yielding a pair of scores: s^i_{anc} denotes the anchor score in the i-th comparison, and s_i denotes the corresponding score of τ_i. To establish the relative ranking within the group, we first compute the anchor's average score $s_{anc} = \frac{1}{N-1}\sum_{i=1}^{N-1} s^i_{anc}$ and then derive the final ranking based on the set of group scores formed by $s_i,\ i = \{1, 2, ..., N-1\}$ and s_{anc}. Although this topology attains linear computational complexity $\mathcal{O}(N)$, it suffers from a loss of resolution. It effectively quantifies the extent to which a sample outperforms the anchor, but fails to capture subtle differences between two exploratory samples, which may lead to ambiguity in ranking among suboptimal solutions.

4.3 Seeded Single-Elimination

To bridge the trade-off between computational efficiency and ranking resolution, we propose a hybrid topology: Seeded Single-Elimination. This approach operates in two distinct phases:

(1) Seeding Phase. We first employ the anchor-based ranking mechanism (described in Section 4.2) to compute a preliminary score for each trajectory, and then sequentially assign a seed ranking s^i_{seed} to obtain a low-bias initial ordering. This initialization is critical for mitigating premature collisions, where high-quality trajectories might otherwise meet and eliminate each other in early rounds.

(2) Elimination Phase. We construct a binary tournament tree in which match-ups are arranged according to seed rankings (*e.g.*, pairing the highest seed with the lowest: seed 1 vs. seed N). In each match, the winner advances while the loser is eliminated:

$$\tau_{\text{win}} = \text{argmax}_{\tau \in \tau_i, \tau_j}(s_i, s_j) \tag{5}$$

The final ranking is primarily determined by the depth of survival within the tournament bracket. For trajectories eliminated in the same round (*e.g.*, quarter-finals), intra-tier ties are further ranked using their accumulated average scores from previous matches. This topology preserves linear complexity $\mathcal{O}(N)$, specifically, requiring $N-1$ comparisons for seeding and $N-1$ for the tournament. Crucially, by leveraging high-quality priors from the seeding phase to guide the tournament structure, this method yields an accurate estimate of relative rankings, ensuring that strategy updates are driven by genuinely superior reasoning trajectories.

4.4 Double-Elimination Tournament

We further investigate the Double-Elimination Tournament topology for group ranking estimation. Unlike the single-elimination format, this structure incorporates a losers' bracket, so that a trajectory is eliminated only after sustaining two defeats. The ranking criteria mirror those of Seeded Single-Elimination, relying on advancement depth and accumulated average scores. To maintain a computational budget comparable to Seeded Single-Elimination ($\approx 2N$ comparisons), we initialize this format with random seeding rather than the anchor-based ranking mechanism. Although this topology is in principle more robust to isolated upsets, empirical results indicate that, without high-quality initial seeds, its ranking fidelity falls short of that achieved by Seeded Single-Elimination.

4.5 Swiss-System Tournament

We also evaluate the Swiss-System Tournament, a non-elimination format with dynamic pairing. In each round, trajectories with identical win–loss records are matched against one another (*e.g.*, a "1–0" candidate competes against another "1–0" candidate). All trajectories participate in a fixed number of rounds ($K \approx \log_2 N$), with each round comprising $N/2$ matches. Final rankings are determined

by a composite metric consisting of total wins and the Buchholz score (the sum of wins achieved by a trajectory's past opponents). This topology incurs a computational complexity of $\mathcal{O}(N \log N)$.

4.6 Ranking-Based Policy Optimization

Irrespective of the underlying tournament topology, ArenaRL produces a relative ranking $\text{Rank}(\tau_i) \in \{0, \ldots, N-1\}$ for each trajectory in the group, where 0 denotes the highest rank. To enable stable optimization, we convert these discrete ranks into normalized advantage signals. We first map the ranks to quantile-based rewards:

$$r_i = 1 - \frac{\text{Rank}(\tau_i)}{N-1}. \tag{6}$$

We then compute the standardized advantage A_i within the group:

$$A_i = \frac{r_i - \mu_r}{\sigma_r + \epsilon}, \tag{7}$$

where μ_r and σ_r denote the mean and standard deviation of the rank-based rewards $\{r_1, \ldots, r_N\}$, respectively. Finally, we optimize the policy by maximizing the following objective function, which incorporates a KL-divergence penalty to discourage excessive deviation from the reference policy π_{ref}:

$$\begin{aligned}\mathcal{L}_{\text{ArenaRL}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, \mathcal{G} \sim \pi_\theta} \Bigg[\frac{1}{N} \sum_{i=1}^{N} \Bigg(\min \left(\frac{\pi_\theta(\tau_i \mid x)}{\pi_{\text{old}}(\tau_i \mid x)} A_i, \text{clip} \left(\frac{\pi_\theta(\tau_i \mid x)}{\pi_{\text{old}}(\tau_i \mid x)}, 1-\epsilon, 1+\epsilon \right) A_i \right) \\ - \beta \, \mathbb{D}_{\text{KL}} \big(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x) \big) \Bigg) \Bigg].\end{aligned} \tag{8}$$

By transforming relative quality ranking within trajectory groups into a stable advantage signal, ArenaRL effectively drives the policy toward increasingly strong reasoning and planning behaviors on open-ended tasks.

5 Benchmarking Open-Ended Agency

In this section, we introduce Open-Travel and Open-DeepResearch, two benchmarks constructed from realistic business scenarios and designed to comprehensively evaluate LLMs on open-ended agentic tasks. For each domain, we define domain-specific data, including shared tools, domain policy texts, and task instances. Each domain is instantiated through a three-stage construction pipeline (as shown in Figure 3) that systematically decomposes real-world scenarios and incorporates expert annotation and rigorous validation.

Stage I: Benchmark Data Collection This phase establishes a rigorous evaluation foundation by curating high-quality, scenario-specific queries and generating reference trajectories to facilitate robust comparative analysis.

- **Open-ended query construction.** We begin by abstracting real-world application scenarios (such as travel planning and open-ended deep research) and collecting a corpus of authentic user queries for each scenario. Domain experts then perform multiple rounds of filtering and refinement to obtain a set of queries that are semantically precise and succinctly formulated. Finally, we select 50 queries for each of the five Open-Travel subtasks and 100 queries for Open-DeepResearch as the benchmark test sets.
- **Baseline trajectory construction.** To obtain a reference baseline for subsequent pairwise comparison and win-rate evaluation (*i.e.*, the basic agent capability level), we adopt high-performing closed-source models as the base models to generate complete tool-use trajectories and the corresponding open-ended answers.

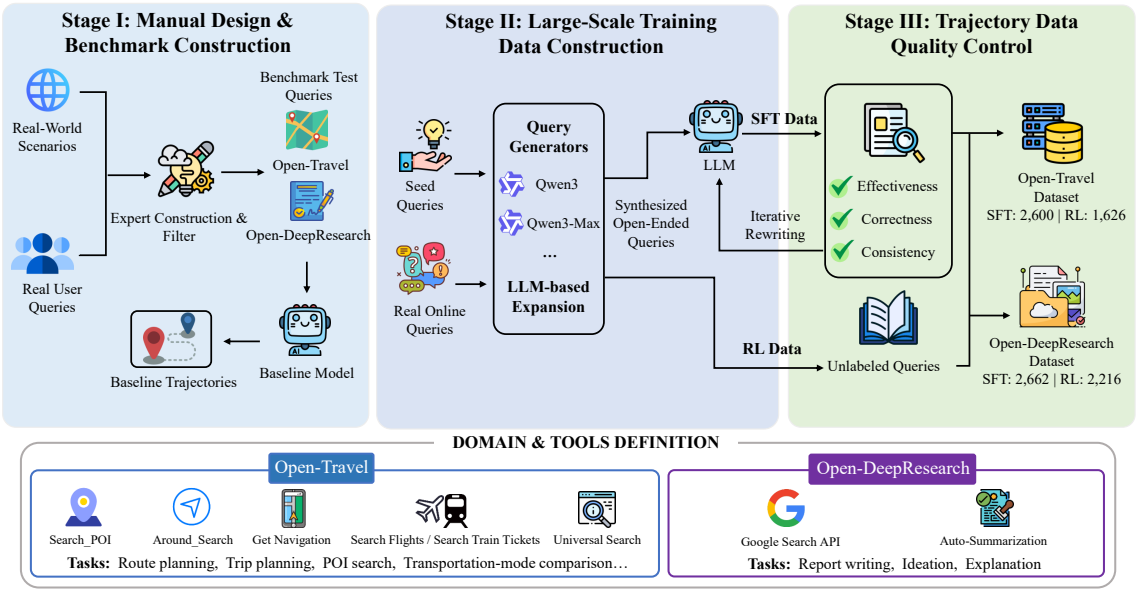

Figure 3: The construction process of Open-Travel and Open-DeepResearch benchmarks.

Stage II: Large-Scale Training Data Construction After fixing the benchmark, we first manually construct a small set of representative seed queries tailored to the requirements of different sub-tasks in real business scenarios. We then use multiple LLMs with diverse styles (*e.g.*, Qwen3 Yang et al. (2025), Qwen3-Max Bai et al. (2023)) as "query generators" to synthesize large-scale, multi-scenario open-ended queries. The resulting training dataset thus contains both complete queries collected from real business workflows and diversified queries produced via LLM-based expansion.

Consistent with Stage I, we employ high-performing closed-source models as the base model to generate large-scale tool-use trajectories, which are used as cold-start data for SFT. The remaining queries, without explicit supervised trajectories, are reserved for subsequent RL.

Stage III: Trajectory Data Quality Control We first feed the complete trajectory dataset into a rule-augmented LLM-based quality inspection module, which evaluates each trajectory along three dimensions: effectiveness of open-ended tool usage, correctness of conversational content, and consistency of the final answers. This process filters out trajectories with formatting or logical errors. For trajectories that fail to meet the criteria, we iteratively rewrite and refine them until they pass the LLM-based quality inspection.

5.1 Domains

Using the above pipeline, we modularly construct two domains: Open-Travel and Open-DeepResearch. In the following sections, we provide a brief description of the policies for each domain.

Open-Travel. In the Open-Travel domain, the agent is required to help users accomplish the following five types of itinerary planning subtasks, which jointly emphasize multi-constraint reasoning, multi-tool coordination, and personalized preferences:

- Route planning with multiple specified waypoints (defined as the **Direction** subtask);
- One-day trip planning in a single city (denoted as (defined as the **1-Day** subtask);
- Transportation-mode comparison (defined as the **Compare** subtask);

Table 1: Statistics of the constructed Open-Travel and Open-DeepResearch benchmarks.

Dataset	Training Set		Test Samples	Language	Domain
	SFT	RL			
Open-Travel	2,600	1,626	250	Chinese	Travel Planning
Open-DeepResearch	2,662	2,216	100	Chinese / English	General
Total	**5,262**	**3,842**	**350**	Chinese / English	–

- Nearby point-of-interest (POI) search (defined as the **Search** subtask);
- Multi-day trip planning, evaluated as a generalization task and excluded from the SFT training data (defined as the **M-Day** subtask).

These rules are further intertwined with user-specific constraints (*e.g.*, budget limits, time windows, traveling parties, and preference profiles), resulting in challenging reasoning tasks for the agent.

Open-DeepResearch. In the Open-DeepResearch domain, the agent is required to assist users in conducting multi-turn search, reading, synthesis, and generation, and ultimately produce an open-ended answer. The open-ended summarization and generation tasks in this domain mainly fall into the following categories:

- Assisting users in writing open-ended technical documents (*e.g.*, reports, design documents, or survey-style overviews);
- Helping users ideate, expand, or refine research topics, solution plans, or content outlines;
- Providing concise yet informative explanations, overviews, or summaries of complex concepts, systems, or domains.

5.2 Dataset Statistics and Analysis

As shown in Table 1, the final Open-Travel and Open-DeepResearch datasets are constructed at a reasonable scale with high diversity, providing the research community with a practical and representative benchmark for evaluating open-ended agentic reasoning and tool-use capabilities.

5.2.1 Dataset Scale

Training Set. Open-Travel contains 2,600 SFT samples and 1,626 RL samples, while Open-DeepResearch contains 2,662 SFT samples and 2,216 RL samples. The SFT data are mainly used to help the model acquire basic tool-calling formats, intent understanding, and multi-step reasoning patterns. The RL query samples are then used to further elicit and optimize the model's open-ended agentic behaviors under realistic constraints.

Test Set. We construct a high-quality test set for leaderboard-style evaluation, consisting of 250 samples from Open-Travel and 100 samples from Open-DeepResearch. All test samples are manually checked to ensure representative clarity, diversity, and difficulty.

5.2.2 Category Coverage

The dataset spans a wide spectrum of functional categories. Beyond the travel-planning domain, it also covers areas such as sports, medicine, and a variety of other everyday and professional scenarios. This broad topical distribution allows us to evaluate models not only on their specialized performance in specific domains (*e.g.*, travel planning), but also on their overall competence as general-purpose open agents.

5.3 Evaluation

We adopt an LLM-as-a-judge evaluation paradigm, and use two strong proprietary models as dual judges to score both the reasoning trajectories and the final answers.

Open-Travel. For Open-Travel, we evaluate models on the carefully curated Open-Travel test set. For each test sample, we independently invoke two powerful closed-source LLM judges from different model families to perform pairwise evaluation. Each judge compares the candidate agent's output with the baseline output and assigns scores along multiple dimensions (*e.g.*, answer correctness, consistency with the reasoning trajectory, etc.). Based on these judgments, we compute the win rate for each evaluation criterion, defined as the proportion of non-tied cases where the candidate output is preferred over the baseline. We then average the win rates obtained from the two judges and use this averaged value as the final performance metric for each Open-Travel subtask.

Open-DeepResearch. For Open-DeepResearch, we adopt the same evaluation protocol as used for Open-Travel on the Open-DeepResearch test set. For each sample, two closed-source LLM judges from different model families are independently employed to evaluate both the reasoning trajectory and the final answer jointly.

Notably, due to the long-context nature of DeepResearch tasks, models may occasionally experience context overflow, resulting in the inability to generate valid final answers. To account for this issue, we additionally report the valid generation rate (Val. %) for each model, defined as the proportion of test cases in which a valid answer is successfully generated over the entire benchmark.

Furthermore, for each evaluation criterion, we compute the candidate model's win rate against the baseline conditioned on valid generations, *i.e.*, the proportion of cases with valid outputs in which the candidate model is preferred over the baseline. We then aggregate the per-criterion win rates as well as the cross-criterion average win rate. As in Open-Travel, the final evaluation metric is obtained by averaging the scores produced by the two judge models.

6 Experiments

To comprehensively evaluate the effectiveness of ArenaRL, we first report the performance of the five tournament topologies introduced in Section 4, thereby empirically justifying our selection of the Seeded Single-Elimination scheme. Building upon this optimal topology, we benchmark ArenaRL against strong baseline methods on our proposed Open-Travel and Open-DeepResearch datasets. Furthermore, recognizing the shared characteristics of open-ended problems, we extend our evaluation to standard open-ended writing tasks using three public benchmarks. Finally, to assess the robustness and practical applicability of ArenaRL in real-world settings, we conduct additional experiments on real business data derived from the Amap (Gaode Map) ecosystem.

6.1 Experimental Settings

Baselines. We evaluate ArenaRL against two categories of baselines. First, we benchmark against four closed-source models, including GPT-4o Achiam et al. (2023), Grok-4 xAI (2025), Gemini-2.5-pro Team et al. (2023), and Claude-3.7-Sonnet Anthropic (2023). Second, we compare with representative reinforcement learning algorithms, specifically GRPO Shao et al. (2024) and GSPO Zheng et al. (2025). For these RL algorithm baselines, we employ the standard LLM-as-Judge setting, obtaining rewards through pointwise scoring. To maintain fairness, these baseline algorithms utilize the exact same judge models and evaluation rubrics as ArenaRL, and only evaluate the answer portion.

Training Guideline. Our experiments strictly follow the common "Cold-start → RL" paradigm to mitigate reward collapse during the initial RL exploration phase.

Table 2: Performance comparison of the five tournament topologies on the Open-Travel benchmark.

Topology	Comparison Cost	Open-Travel					Mean
		Direction	Search	Compare	1-Day	M-Day	
SFT	-	10.6	29.7	14.1	20.4	7.1	16.4
Anchor-Based Ranking	$N-1$	18.0	41.3	30.9	31.1	17.6	27.8
Swiss-System	$N \log N$	20.9	43.0	27.9	38.6	11.1	28.3
Double-Elimination	$2N-2$	12.6	52.4	**33.7**	**39.9**	12.3	30.2
Seeded Single-Elimination	$2N-2$	16.9	**69.9**	22.9	34.9	18.1	32.5
Round-Robin (Upper Bound)	$N(N-1)/2$	**23.3**	66.3	23.6	32.1	**19.0**	**32.9**

1. **Cold-start phase.** We utilize Qwen3-8B-Base Yang et al. (2025) as the backbone model. For the open-ended agent tasks (Open-Travel and Open-DeepResearch), the base model is fine-tuned on their respective SFT datasets to acquire fundamental tool-use and planning capabilities. For open-ended writing tasks, we randomly sample 10k examples from the DeepWriting-20K Wang et al. (2025a) dataset for supervised fine-tuning.
2. **RL phase.** For the open-ended agent tasks, we train on their corresponding RL splits. For open-ended writing, we utilize 10k examples from the DeepWriting-20K dataset (excluding those used for SFT) to conduct reinforcement learning.

Evaluation Metrics. For open-ended agent tasks, we perform pairwise evaluations against the baseline trajectories in our benchmark using the multi-dimensional criteria defined in Section 5, and compute the corresponding win rates. For the Open-Travel tasks, we report the win rate on each of the five subtasks as well as the average win rate.

For the Open-DeepResearch tasks, we report the valid generation rate (Val. %). And within the subset of valid generations, the win rates of the candidate model against the baseline under each of the seven evaluation rubrics, along with the final averaged win rate. Specifically, these rubrics capture complementary aspects of open-ended research capability: **Framework** (*Frm.*), assessing the structural completeness and logical coherence of the initial research plan; **Tool Usage** (*Tool.*), evaluating the appropriateness and efficiency of tool invocations and their alignment with the research workflow; **Coverage** (*Cov.*), measuring whether the retrieved information sufficiently covers the user's requirements; **Relevance** (*Rel.*), assessing how well the response addresses all user queries and constraints; **Accuracy** (*Acc.*), evaluating the factual correctness and internal consistency of the content; **Depth** (*Dep.*), measuring the level of analytical depth and coherence of the reasoning process; and **Clarity** (*Cla.*), assessing the organization, readability, and practical usability of the final output. The complete task prompt for the judge models is shown in Figure 6.

For open-ended writing, we adopt three complementary benchmarks for a comprehensive assessment: WritingBench Wu et al. (2025b), HelloBench Que et al. (2024), and LongBench-write Bai et al. (2024). Considering the subjective nature of open-ended generation, and following established protocols Wang et al. (2025a), we employ the LLM-as-judge approach to score the generative quality of different models.

6.2 Tournament Topology Analysis

Table 2 presents a systematic comparison of different tournament topologies under a unified RL configuration (group size $N = 8$, number of groups $K = 8$). The results indicate that the proposed Seeded Single-Elimination scheme achieves the best trade-off between efficiency and performance. Specifically, it attains an average win rate of 32.5%, which is comparable to the "gold standard" performance of 32.9% established by the computationally expensive Round-Robin tournament, while requiring only $\mathcal{O}(N)$ pairwise comparisons. In contrast, the Swiss Round and Double-Elimination formats fail to deliver comparable performance gains, due either to the lack of an effective initial prior or insufficient comparison depth. Notably, Seeded Single-Elimination even outperforms

Table 3: Performance comparison on Open-Travel and Open-DeepResearch benchmarks.

Method	Open-Travel						Open-DeepResearch							
	Direction	Search	Compare	1-Day	M-Day	**Mean**	Frm.	Tool.	Cov.	Rel.	Acc.	Dep.	Cla.	**Mean (Val. %)**
Closed-source Models														
GPT-4o	2.4	5.0	3.1	1.6	0.7	2.6	5.1	24.4	21.0	9.1	12.5	2.3	10.8	12.2 (88.0)
Grok-4	17.0	21.3	9.7	24.7	11.3	16.8	33.7	36.8	43.4	36.8	39.2	36.1	17.5	34.8 (83.0)
Gemini-2.5-pro	8.6	12.5	7.4	11.9	12.4	10.6	15.8	19.0	17.9	32.6	28.3	45.7	38.6	28.3 (92.0)
Claude-3.7-Sonnet	18.6	59.6	14.7	43.6	21.3	31.6	10.1	13.5	22.5	23.6	19.7	27.0	17.4	19.1 (89.0)
Fine-tuning & RL														
SFT	10.6	29.7	14.1	20.4	7.1	16.4	14.1	20.3	23.4	14.1	15.6	15.6	14.1	16.7 (32.0)
GRPO	11.0	26.3	14.3	21.9	8.6	16.4	20.6	35.3	35.3	23.5	23.5	26.5	11.8	25.2 (17.0)
GSPO	10.0	30.6	13.1	21.1	11.4	17.2	23.8	33.3	40.5	16.7	21.4	31.0	9.5	25.2 (21.0)
ArenaRL	**32.1**	**66.1**	**31.7**	**58.0**	**21.0**	**41.8**	**62.6**	**77.3**	**78.8**	**57.1**	**55.6**	**57.1**	**61.6**	**64.3 (99.0)**

Table 4: Performance comparison on open-ended writing task across three public benchmarks: WritingBench, HelloBench, and LongBench-write.

Method	WritingBench						HelloBench			LongBench	Mean
	WB-A	WB-B	WB-C	WB-D	WB-E	WB-F	QA	Summ.	Heur.	Quality	
Closed-source Models											
GPT-4o	67.90	66.34	68.56	69.95	70.70	72.17	81.03	84.26	89.14	90.43	76.05
Grok-4	80.32	78.65	79.75	81.46	81.19	80.92	88.42	85.58	94.65	96.52	84.75
Gemini-2.5-pro	80.89	80.39	82.49	84.33	83.53	82.61	85.67	82.43	93.79	98.69	85.48
Claude-3.7-Sonnet	68.36	66.53	68.70	70.34	71.42	71.47	80.81	74.68	95.82	98.34	76.65
Fine-tuning & RL											
SFT	70.71	69.36	67.88	63.72	69.69	70.64	78.44	63.42	82.35	85.52	72.17
GRPO	71.62	71.18	68.67	66.84	72.56	70.33	79.09	64.94	83.76	86.96	73.60
GSPO	71.56	70.70	68.87	66.08	72.29	69.83	**80.06**	63.97	81.75	85.21	73.03
ArenaRL	**78.14**	**77.70**	**77.58**	**75.02**	**79.35**	**77.16**	79.11	**73.82**	**91.33**	**93.78**	**80.30**

Round-Robin on the *Search* and *1-Day* subtasks. This observation suggests that the anchor-based seeding mechanism effectively filters out noise and prevents high-quality candidates from being adversely affected by random matching fluctuations in the early stages. Based on these findings, we adopt Seeded Single-Elimination as the primary tournament topology in subsequent experiments and conduct further comparison with the remaining strong baselines.

6.3 Main Results

As shown in Table 3, ArenaRL demonstrates strong performance across both open-ended agent benchmarks, outperforming four powerful closed-source models. On the Open-Travel benchmark, ArenaRL achieves an average win rate of 41.8%, substantially outperforming GRPO (16.4%) and GSPO (17.2%). In the Open-DeepResearch benchmark, ArenaRL not only achieves a win rate of 64.3%, but also attains a valid generation rate (Val. %) of 99%. In sharp contrast, the baseline methods perform poorly in terms of task completion, with the SFT model achieving a valid generation rate of only 32%. We attribute this gap primarily to the inherently high token consumption required by deepresearch tasks, together with the prevalence of long-horizon samples in the SFT training data, which jointly lead to frequent context overflows. Notably, although GRPO and GSPO slightly improve the average win rate, their valid generation rates are inferior to that of the SFT baseline. These results highlight a key limitation of standard pointwise reward schemes such as GRPO and GSPO: for long-horizon tasks involving complex tool use, assigning a scalar score to a single trajectory often fails to capture fine-grained policy improvements, and tends to be susceptible to spurious advantages such as length bias. In contrast, ArenaRL's comparison-based reward signal

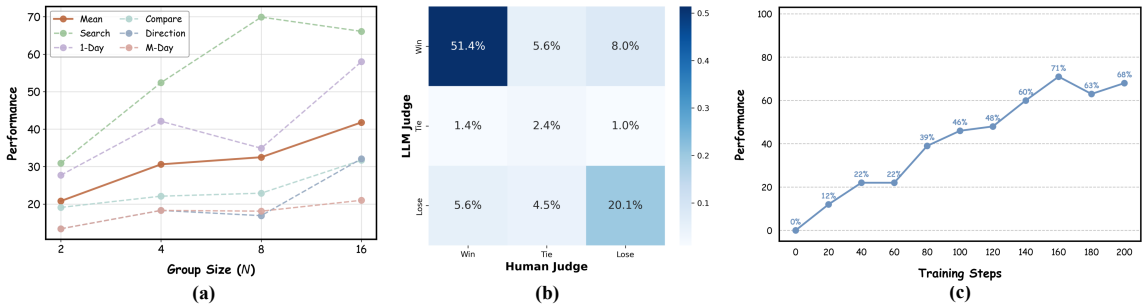

Figure 4: **(a)** The impact of Group Size N on performance of Open-Travel benchmark. **(b)** The consistency between LLM and human evaluations. **(c)** The performance trend of ArenaRL in training Qwen3-8b via direct RL without cold start.

provides more discriminative gradient directions, effectively steering policy evolution toward more robust planning and reasoning capabilities within a vast search space.

Table 4 further validates the generality of ArenaRL on standard open-ended writing tasks. Across three benchmarks, ArenaRL maintains a substantial lead in overall average score, outperforming GRPO by 6.70% and GSPO by 7.27%. In addition, ArenaRL surpasses two strong closed-source models, GPT-4o and Claude-3.7-Sonnet, further highlighting its superiority. Although ArenaRL is slightly inferior to GSPO on the HelloBench-QA subtask, this category of tasks is largely constrained by the model's inherent knowledge, and ArenaRL remains highly competitive under this limitation. Across various open-ended writing scenarios, ArenaRL's comprehensive improvement indicate that ArenaRL is not only suitable for tool-augmented agents, but can also systematically enhance the model's reasoning and expressive capabilities, thereby making it applicable to a broader range of open-ended generation tasks.

6.4 Further Analysis

Impact of Group Size N. We conduct an ablation study with a set group size $N \in \{2, 4, 8, 16\}$ to further investigate the scalability properties of ArenaRL, as shown in Figure 4 (a). The results show a clear monotonic improvement in model performance as the group size increases. Notably, even under the smallest configuration $N = 2$, ArenaRL achieves an average win rate of 20.8%, outperforming the SFT baseline (16.4%). This confirms that even the most basic pairwise comparison setting can provide effective optimization gradients. The most pronounced performance gain occurs when N is scaled up to 16, where the average win rate jumps to 41.8%. This effect is particularly striking on the challenging 1-Day planning task, where the score jumps from 34.9% at $N = 8$ to 58.0%. These substantial improvements indicate that, for complex reasoning tasks, enlarging the candidate pool effectively broadens the exploration space, dramatically increasing the likelihood of discovering high-quality trajectories and thereby enabling the model to learn from stronger examples.

Assessment of Consistency. To assess the reliability of the LLM-based evaluation mechanism, we analyzed the consistency between LLM and human evaluations on the Open-Travel and Open-DeepResearch benchmarks using a confusion matrix (see Figure 4 (b)). The results show that most evaluation outcomes are concentrated along the diagonal, with an overall agreement rate of 73.9%. This relatively high level of consistency suggests that ArenaRL's performance gains do not simply stem from overfitting to the preferences of the specific judge model used during the RL phase, but instead reflect improvements that are broadly aligned with human assessments.

Direct RL Training without Cold Start. To further test the robustness of ArenaRL, we bypassed the cold-start phase on the Open-Travel task and directly employed the Qwen3-8B model for ArenaRL

training, reporting its performance on the *Search* sub-task. As shown in Figure 4 (c), the model's performance exhibited a sustained and stable upward trend. At the initial step 0, the model scored 0, indicating that the generic model was initially incapable of handling such complex travel planning tasks. However, as RL training steps increase, the model rapidly acquired the corresponding tool invocation capabilities. Ultimately, it achieves a peak score of 71% at step 160. This outcome indicates that ArenaRL's intra-group relative ranking mechanism sensitively captures effective optimization directions. Even when initial output quality is extremely low, it reliably provides gradients for policy refinement. This effectively mitigates RL's cold-start problem, demonstrating ArenaRL's capacity for self-evolution from scratch in scenarios lacking costly SFT annotated data.

Case Study. Figures F.1 and F.2 present a representative comparative case from the Open-Travel benchmark. This example involves a complex travel request with multiple hard constraints, including time, destination, and budget. The baseline SFT model exhibits a restatement tendency in its chain-of-thought, and its reasoning trajectory fails to align with the user's intent, often overlooking specific constraints and providing only generic suggestions. In contrast, the model optimized with ArenaRL demonstrates strong strategic planning capabilities. It proactively retrieves information about multiple target attractions, performs logically coherent route planning, and ultimately produces a persuasive, personalized itinerary. This substantial improvement in reasoning patterns powerfully demonstrates that our tournament-based ranking mechanism effectively incentivizes the model to explore and retain superior planning strategies.

6.5 Application in Real-world Business Scenarios

To further verify the robustness and practicality of our proposed ArenaRL algorithm, we conducted experiments on real-world business data derived from the Amap (Gaode Map) ecosystem. The evaluation was divided into two distinct categories based on the nature of the user queries: quantifiable POI search tasks and complex open-ended tasks.

Performance on Deterministic POI Search. In POI search scenarios characterized by explicit evaluation metrics, our ArenaRL-tuned model significantly outperformed the baseline, registering a 75% to 83% gain in search accuracy. These results confirm the model's ability to navigate rigid constraints and specific ranking criteria. Furthermore, this demonstrates that the tournament-based ranking mechanism of ArenaRL is highly effective at distinguishing subtle nuances across varied high-quality trajectories, ensuring robust optimization performance in deterministic task settings.

Performance on Open-ended Planning Tasks. We extended our evaluation to complex, open-ended travel planning tasks that require multi-step reasoning and tool invocation. These scenarios include queries with vague intents and ambiance preferences (*e.g.*, 'Find a quiet bar near the Bund with a river-view terrace for a date, open after 10 PM') as well as complex cross-city logistics requiring multi-objective trade-offs (*e.g.*, 'Depart Beijing West at 18:30, arrive in Tianjin by 22:00; minimize cost and transfers due to heavy luggage'). On these tasks, the core business metric rose significantly from 69% to 80%. We observed substantial and consistent gains throughout the training process, reflecting the model's enhanced capability to interpret ambiguous intents and satisfy multiple constraints. These results indicate that the efficient planning capabilities acquired via ArenaRL effectively transfer to practical applications, significantly improving user intent alignment and response quality in complex service scenarios.

7 Conclusion

In this paper, we propose ArenaRL, a novel reinforcement learning framework that shifts the training paradigm for open-ended agents from pointwise scalar scoring to intra-group relative ranking. Specifically, we introduce a process-aware pairwise evaluation mechanism and systematically investigate five tournament topologies. Our findings reveal that the seeded single-elimination topology strikes a favorable balance between advantage estimation accuracy and computational

efficiency. Extensive evaluations across travel planning, deep research, and open-ended writing tasks demonstrate that this competition-driven evolutionary paradigm not only provides robust advantage signals, but also fundamentally incentivizes agents to perform efficient reasoning and planning. As future work, we will explore how to efficiently extend ArenaRL to multimodal agent settings, further enhancing the generality of our framework.

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.

Anthropic. Introducing Claude, 2023. URL https://www.anthropic.com/index/introducing-claude.

Esteban Garces Arias, Meimingwei Li, Christian Heumann, and Matthias Aßenmacher. Decoding decoded: Understanding hyperparameter effects in open-ended text generation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 9992–10020, 2025.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longwriter: Unleashing 10,000+ word generation from long context llms. *arXiv preprint arXiv:2408.07055*, 2024.

João Coelho, Jingjie Ning, Jingyuan He, Kangrui Mao, Abhijay Paladugu, Pranav Setlur, Jiahe Jin, Jamie Callan, João Magalhães, Bruno Martins, et al. Deepresearchgym: A free, transparent, and reproducible evaluation sandbox for deep research. *arXiv preprint arXiv:2505.19253*, 2025.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*, 2025.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. Deepresearch bench: A comprehensive benchmark for deep research agents. *arXiv preprint arXiv:2506.11763*, 2025.

Johannes Fürnkranz and Eyke Hüllermeier. Preference learning and ranking by pairwise comparison. In *Preference learning*, pp. 65–82. Springer, 2010.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Wei He, Yueqing Sun, Hongyan Hao, Xueyuan Hao, Zhikang Xia, Qi Gu, Chengcheng Han, Dengchang Zhao, Hui Su, Kefeng Zhang, et al. Vitabench: Benchmarking llm agents with versatile interactive tasks in real-world applications. *arXiv preprint arXiv:2509.26490*, 2025.

Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Tianfu Wang, Zhengyu Chen, Nicholas Jing Yuan, Jianxun Lian, Kaize Ding, and Hui Xiong. Explaining length bias in llm-based preference evaluations. *arXiv preprint arXiv:2407.01085*, 2024.

Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, et al. Reinforcement learning with rubric anchors. *arXiv preprint arXiv:2508.12790*, 2025.

Ruipeng Jia, Yunyi Yang, Yongbo Gai, Kai Luo, Shihao Huang, Jianhe Lin, Xiaoxi Jiang, and Guanjun Jiang. Writing-zero: Bridge the gap between non-verifiable tasks and verifiable rewards. *arXiv e-prints*, pp. arXiv–2506, 2025.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*, 2023.

Xiaoxi Li, Wenxiang Jiao, Jiarui Jin, Guanting Dong, Jiajie Jin, Yinuo Wang, Hao Wang, Yutao Zhu, Ji-Rong Wen, Yuan Lu, et al. Deepagent: A general reasoning agent with scalable toolsets. *arXiv preprint arXiv:2510.21618*, 2025a.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*, 2025b.

Zijian Li, Xin Guan, Bo Zhang, Shen Huang, Houquan Zhou, Shaopeng Lai, Ming Yan, Yong Jiang, Pengjun Xie, Fei Huang, et al. Webweaver: Structuring web-scale evidence with dynamic outlines for open-ended deep research. *arXiv preprint arXiv:2509.13312*, 2025c.

Tianci Liu, Ran Xu, Tony Yu, Ilgee Hong, Carl Yang, Tuo Zhao, and Haoyu Wang. Openrubrics: Towards scalable synthetic rubric generation for reward modeling and llm alignment. *arXiv preprint arXiv:2510.07743*, 2025.

Yansong Ning, Rui Liu, Jun Wang, Kai Chen, Wei Li, Jun Fang, Kan Zheng, Naiqiang Tan, and Hao Liu. Deeptravel: An end-to-end agentic reinforcement learning framework for autonomous travel planning agents. *arXiv preprint arXiv:2509.21842*, 2025.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

Haoran Que, Feiyu Duan, Liqun He, Yutao Mou, Wangchunshu Zhou, Jiaheng Liu, Wenge Rong, Zekun Moore Wang, Jian Yang, Ge Zhang, et al. Hellobench: Evaluating long text generation capabilities of large language models. *arXiv preprint arXiv:2409.16191*, 2024.

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.

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.

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.

Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, et al. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*, 2025.

Vijay Viswanathan, Yanchao Sun, Shuang Ma, Xiang Kong, Meng Cao, Graham Neubig, and Tongshuang Wu. Checklists are better than reward models for aligning language models. *arXiv preprint arXiv:2507.18624*, 2025.

Haozhe Wang, Haoran Que, Qixin Xu, Minghao Liu, Wangchunshu Zhou, Jiazhan Feng, Wanjun Zhong, Wei Ye, Tong Yang, Wenhao Huang, et al. Reverse-engineered reasoning for open-ended generation. *arXiv preprint arXiv:2509.06160*, 2025a.

Yibin Wang, Zhimin Li, Yuhang Zang, Yujie Zhou, Jiazi Bu, Chunyu Wang, Qinglin Lu, Cheng Jin, and Jiaqi Wang. Pref-grpo: Pairwise preference reward-based grpo for stable text-to-image reinforcement learning. *arXiv preprint arXiv:2508.20751*, 2025b.

Yuanchen Wu, Ke Yan, Shouhong Ding, Ziyin Zhou, and Xiaoqiang Li. Towards rationale-answer alignment of lvlms via self-rationale calibration. *arXiv preprint arXiv:2509.13919*, 2025a.

Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Zijia Wang, Ji Zhang, Mengyue Wu, Qin Jin, et al. Writingbench: A comprehensive benchmark for generative writing. *arXiv preprint arXiv:2503.05244*, 2025b.

xAI. Grok 4, 2025. URL https://x.ai/news/grok-4.

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

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

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.

Fanrui Zhang, Qiang Zhang, Sizhuo Zhou, Jianwen Sun, Chuanhao Li, Jiaxin Ai, Yukang Feng, Yujie Zhang, Wenjie Li, Zizhen Li, et al. Code-in-the-loop forensics: Agentic tool use for image forgery detection. *arXiv preprint arXiv:2512.16300*, 2025.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Zilin Zhu, Chengxing Xie, Xin Lv, and slime Contributors. slime: An llm post-training framework for rl scaling. https://github.com/THUDM/slime, 2025. GitHub repository. Corresponding author: Xin Lv.

A Implementation Details.

Cold-start phase. We employ the TRL framework optimized with DeepSpeed ZeRO-3. The model is trained for 3 epochs on 32 $\times$ NVIDIA H20 GPUs, utilizing a learning rate of 2×10^{-5} and a batch size of 1 per GPU.

RL phase. we implement ArenaRL based the Slime Zhu et al. (2025) framework. To focus optimization on reasoning quality, tokens corresponding to environmental feedback are masked out from the loss computation. Regarding hyperparameters, we set the group size $N = 16$ and the number of groups $K = 8$ for Open-Travel and open-ended writing tasks. For Open-DeepResearch, we adjust $N = 8$ and $K = 4$ to enhance training efficiency. Across all RL experiments, we employ the Adam optimizer with a learning rate of 1×10^{-6}. Furthermore, we utilize Qwen3-Max as the arena judge during training, leveraging its superior reasoning capabilities to provide high-fidelity pairwise comparison signals, with its prompt detailed in Appendix E. Besides, the RL training is conducted on 8 $\times$ H20 GPUs.

Evaluation For the Open-Travel and Open-DeepResearch benchmarks, we invoke Qwen3-Max and Claude-4-Sonnet Anthropic (2023) as evaluation model, comparing and scoring the candidate agent's output against the baseline output for each test sample.

B Writing Benchmarks

To ensure the comprehensiveness and multifaceted nature of our evaluation, we adopt three complementary benchmarks: WritingBench Wu et al. (2025b), HelloBench Que et al. (2024), and LongBench-Write Bai et al. (2024).

1. **WritingBench** Wu et al. (2025b) is designed to assess models' domain-specific writing proficiency and controllability across six professional and creative domains: A (Academic & Engineering), B (Finance & Business), C (Politics & Law), D (Literature & Arts), E (Education), and F (Advertising & Marketing). It focuses in particular on the ability to comply with complex, multidimensional constraints.
2. **HelloBench** Que et al. (2024) evaluates model performance on a diverse set of "in-the-wild" tasks derived from real user queries. Our analysis concentrates on three key subsets: HelloBench-QA (open-ended QA), which tests the generation of detailed and nuanced responses; HelloBench-Summ. (text summarization), which evaluates the ability to summarize long-form texts; and HelloBench-Heur. (heuristic text generation), which measures creative reasoning and stylistic fidelity in long-form narrative continuation.
3. **LongBench-write** Bai et al. (2024) is designed to evaluate a model's ability to produce coherent ultra-long texts (*e.g.*, on the order of 10,000 words), enabling us to assess the fundamental capacity to maintain topical consistency and global coherence as output length scales.

C Benchmarking Tool Annotation

In this section, we will outline the tools employed by the Open-DeepResearch and Open-Travel benchmarks respectively.

Open-DeepResearch. We annotate commonly used search tools for open-ended deep research as candidate tools. Specifically, we perform web search via the Google API and integrate a summarization model into the search pipeline to prevent excessive context length. For each retrieved webpage, if the parsed content exceeds 2,500 characters, we invoke the Qwen3-Max Yang et al. (2025) model to automatically summarize the page content.

Open-Travel. We annotate six commonly used tool types for travel planning:

- **Search poi:** This tool is built on Amap's POI search service and retrieves location information via text queries. The input can be a structured address (*e.g.*, "No. 10 Fuyong Street, Wangjing, Chaoyang District, Beijing") or a POI name (*e.g.*, "People's Square"). It returns multiple potentially relevant POIs, including: (1) detailed address; (2) geographic coordinates (the location field, formatted as "longitude, latitude"); and (3) business information (the Business field).
- **Around search:** This tool searches for POIs within a circular area specified by a center point and radius. Users can specify POI types or constrain the returned results using keywords (*e.g.*, "bank"). It returns multiple potentially relevant POIs, each including: (1) detailed address; (2) geographic coordinates (the location field, where longitude and latitude are separated by a comma, with longitude first and latitude second); and (3) business information.
- **Get navigation:** This tool provides integrated route planning based on Amap's navigation service, covering multiple transportation modes such as walking, driving, and public transit. The inputs are the latitude–longitude pairs of the origin and destination, with optional waypoints and a route-planning mode (driving by default). The tool returns a dictionary containing detailed route-planning information.
- **Universal search:** This tool performs general, city-level geographic knowledge search using the search engine. When users pose vague or open-ended questions (*e.g.*, "What are some fun things to do in Beijing?" or "What parks in Shanghai are suitable for family outings?"), It is used to obtain recommendations and travel suggestions. The tool returns the corresponding search results.
- **Search flights:** This tool searches for intercity flight information based on date and city names. When users need to query flights between two cities, they provide the query date and departure city, and the tool returns a list of flights, each including flight number, price, departure/arrival airports, and departure/arrival times.
- **Search train tickets:** This tool searches for intercity train ticket information based on date and city names. When users need to query train tickets between two cities, it returns a list of train options, where each entry includes train ID, price, departure/arrival stations, and times, and explicitly distinguishes between direct and transfer routes.

For these six tools, search poi, around search, and get navigation are implemented using Web service APIs from the Amap open platform[1]. The universal search tool is powered by the Bailian search engine[2]. The outputs of search flights and search train tickets are simulated with the detailed prompts provided in the Appendix E.

D Tournament Algorithm Flow

In Algorithm 1, we present the detailed procedure of the proposed seeded single-elimination tournament, which achieves an excellent trade-off between training efficiency and the accuracy of advantage estimation.

E Prompts

In this section, we will illustrate all the prompts used in our paper.

E.1 Open-Travel Task Prompt

See Figure 5.

[1]https://lbs.amap.com/api/webservice/guide/api-advanced/search

[2]https://bailian.console.aliyun.com/?spm=5176.12818093_47.resourceCenter.1.3dd916d04Ye5xn&tab=app#/mcp-market/detail/WebSearch

E.2 Open-DeepResearch Task Prompt

See Figure 6.

E.3 Open-ended Writing Task Prompt

Prompts for open-ended writing tasks adapt according to the language type of the query, specifically the Chinese prompt (Figure 7) and the English prompt (Figure 8).

E.4 Search Flights Tool Prompt

See Figure 9.

E.5 Search Train Tickets Tool Prompt

See Figure 10.

F Case Study

Figures F.1 and F.2 illustrate the interaction trajectories of the baseline SFT model and the model tuned via ArenaRL within a representative case under the Open-Travel benchmark. These visualizations clearly illustrate that our ArenaRL algorithm substantially enhances the agent's ability to perform complex reasoning and efficient action planning.

Algorithm 1: ArenaRL: Seeded Single-Elimination Advantage Estimation

Input: Trajectory group $\mathcal{G} = \{\tau_{anc}\} \cup \{\tau_1, \ldots, \tau_{N-1}\}$, Arena Judge $\mathcal{J}(\cdot,\cdot)$
Output: Standardized advantages $\mathcal{A} = \{A_1, \ldots, A_N\}$
`// Phase 1: Anchor-Based Seeding (Sec. 4.2)`
Initialize score set $\mathcal{S}_{init} \leftarrow \emptyset$
for $i \leftarrow 1$ **to** $N-1$ **do**
  Obtain scores: $s_i, s^i_{anc} \leftarrow \mathcal{J}(\tau_i, \tau_{anc})$
  $\mathcal{S}_{init} \leftarrow \mathcal{S}_{init} \cup \{(\tau_i, s_i)\}$
end
$s_{anc} \leftarrow \frac{1}{N-1}\sum_{i=1}^{N-1} s^i_{anc}$; $\mathcal{S}_{init} \leftarrow \mathcal{S}_{init} \cup \{(\tau_{anc}, s_{anc})\}$
Sort $\mathcal{G}$ descendingly based on $\mathcal{S}_{init}$ to get ordered seeds: $\mathcal{P} \leftarrow [\tau_1, \ldots, \tau_N]$
Initialize accumulated scores $V[\tau] \leftarrow$ score in $\mathcal{S}_{init}$ for all $\tau \in \mathcal{G}$
Initialize ranking tiers $\mathcal{T} \leftarrow \emptyset$
`// Phase 2: Elimination Tournament (Sec. 4.3)`
Initialize array $\mathcal{B}$ of size N
$idx_{head} \leftarrow 1$; $idx_{tail} \leftarrow N-1$ `// Tail pointer at start of last pair`
for $k \leftarrow 1$ **to** $N/2$ **do**
  $(\tau_{high}, \tau_{low}) \leftarrow (\mathcal{P}[k], \mathcal{P}[N-k+1])$ `// Pair Seed` k `vs` $N-k+1$
  if *k is odd* **then**
    $\mathcal{B}[idx_{head}] \leftarrow \tau_{high}$; $\mathcal{B}[idx_{head}+1] \leftarrow \tau_{low}$
    $idx_{head} \mathrel{+}= 2$ `// Fill from front`
  else
    $\mathcal{B}[idx_{tail}] \leftarrow \tau_{high}$; $\mathcal{B}[idx_{tail}+1] \leftarrow \tau_{low}$
    $idx_{tail} \mathrel{-}= 2$ `// Fill from back`
  end
end
while $|\mathcal{B}| > 1$ **do**
  $\mathcal{W}_{round} \leftarrow \emptyset$; $\mathcal{L}_{round} \leftarrow \emptyset$
  for $k \leftarrow 1$ **to** $|\mathcal{B}|/2$ **do**
    $(\tau_a, \tau_b) \leftarrow (\mathcal{B}[2k-1], \mathcal{B}[2k])$; $(s_a, s_b) \leftarrow \mathcal{J}(\tau_a, \tau_b)$
    $V[\tau_a] \mathrel{+}= s_a$; $V[\tau_b] \mathrel{+}= s_b$
    $(\tau_{win}, \tau_{lose}) \leftarrow (s_a > s_b)?(\tau_a, \tau_b) : (\tau_b, \tau_a)$ `// Determine result`
    $\mathcal{W}_{round} \leftarrow \mathcal{W}_{round} \cup \{\tau_{win}\}$; $\mathcal{L}_{round} \leftarrow \mathcal{L}_{round} \cup \{\tau_{lose}\}$
  end
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathcal{L}_{round}\}$; $\mathcal{B} \leftarrow \mathcal{W}_{round}$ `// Archive losers, promote winners`
end
$\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathcal{B}\}$ `// Champion enters final tier`
`// Phase 3: Rank Assignment (Hierarchical Sorting)`
$k \leftarrow 0$
foreach *Tier* $\in \mathcal{T}$*.reverse()* **do**
  Sort *Tier* descendingly based on accumulated scores V
  Assign Rank$(\tau) \leftarrow k{+}{+}$ for all $\tau \in$ *Tier*
end
`// Phase 4: Advantage Calculation (Sec. 4.6)`
Assign Reward: $r_\tau \leftarrow 1 - \frac{\text{Rank}(\tau)}{N-1}$
Compute statistics μ_r, σ_r from $\{r_\tau\}$
return $A_i \leftarrow \frac{r_{\tau_i} - \mu_r}{\sigma_r + \epsilon}$ for all $\tau_i \in \mathcal{G}$

F.1 Case Study of SFT model

Case Trajectory in Open-Travel Benchmark of SFT model (translated into English)

Query: A two-day weekend hiking trip from Chengdu to Qingcheng Back Mountain for two people, with a focus on natural scenery and photography, and a budget of 600 RMB per person.

Step 1

<think>
The user would like to plan a weekend hiking trip from Chengdu to Qingcheng Back Mountain for two people, with a preference for natural scenery and photography, and a budget of 600 RMB per person. I need to first determine the location of Qingcheng Back Mountain, then plan transportation from Chengdu to that area, as well as hiking routes in Qingcheng Back Mountain.
First, I need to use the search_poi tool to look up the location information of Qingcheng Back Mountain.
</think>
<tool_call>
{"name": "search_poi", "arguments": {"query": "Qingcheng Back Mountain"}}
</tool_call>
<tool_response>
...
</tool_response>

Step 2

<think>
I need to plan a weekend hiking itinerary from Chengdu to Qingcheng Back Mountain for the user, suitable for two people, with a preference for natural scenery and photography, and a budget of 600 RMB per person. I have already obtained the basic location information for Qingcheng Back Mountain, and now need to further understand the transportation options from Chengdu to Qingcheng Back Mountain, as well as the hiking routes in Qingcheng Back Mountain and the surrounding natural scenery and photography spots.
First, I need to find out the transportation options from Chengdu to Qingcheng Back Mountain.
</think>
<tool_call>
{"name": "universal_search", "arguments": {"query": "Transportation options from Chengdu to the Qingcheng Back Mountain"}}
</tool_call>
<tool_response>
...
</tool_response>

Step 3

<think>
I need to plan a weekend hiking itinerary from Chengdu to Qingcheng Back Mountain for the user, suitable for two people, focusing on natural scenery and photography, with a budget of 600 RMB per person. I have already obtained basic information and transportation options for Qingcheng Back Mountain, and now need to further understand the hiking routes and points of interest in Qingcheng Back Mountain in order to design a detailed schedule.
</think>
<tool_call>
{"name": "universal_search", "arguments": {"query": "Hiking Routes and Prime Photography

Spots on Qingcheng Back Mountain"}}
</tool_call>
<tool_response>
...
</tool_response>

Step 4

<think>
I need to plan a weekend hiking itinerary from Chengdu to Qingcheng Back Mountain for the user, with an emphasis on natural scenery and photography, and a budget of 600 RMB per person. I have already obtained the location information and some basic data for both Chengdu and Qingcheng Back Mountain, and now need to further understand the location of Chengdu in order to plan the transportation route.
</think>
<tool_call>
{"name": "search_poi", "arguments": {"query": "Chengdu"}}
</tool_call>
<tool_response>
...
</tool_response>

Step 5

<think>
I need to plan a weekend hiking itinerary from Chengdu to Qingcheng Back Mountain for the user, with a focus on natural scenery and photography, and a budget of 600 RMB per person. I have already obtained the location information of Chengdu and Qingcheng Back Mountain, as well as basic information about transportation and hiking routes. Now I need to design a specific travel plan and related details.
First, I need to determine the specific transportation route from Chengdu to Qingcheng Back Mountain. Based on previous search results, there are several main options for traveling from Chengdu to Qingcheng Back Mountain:
1. Take a high-speed train from Chengdu to Qingchengshan Station, then transfer to a scenic area shuttle bus or taxi to get to the Back Mountain area;
2. Drive directly to Qingcheng Back Mountain.
I will use the get_navigation tool to plan the transportation route from Chengdu to Qingcheng Back Mountain. **</think>**
<tool_call>
{"name": "get_navigation", "arguments": {"start_lon": 104.066301, "start_lat": 30.572961, "end_lon": 103.487136, "end_lat": 30.93071}}
</tool_call>
<tool_response>
...
</tool_response>

Step 6

<answer>
Chengdu to Qingcheng Rear Mountain Hiking Guide (Nature Landscape Photography Trip for Two)
Trip Overview Total budget: 600 RMB per person (1,200 RMB for two) Duration: 2 days over a weekend (depart Saturday, return Sunday) Group size: 2 people Main activities: Hiking, nature landscape photography, leisure sightseeing
Detailed Itinerary Day 1: Chengdu → Qingcheng Rear Mountain Morning: Depart from Chengdu 07:00–08:00 Depart from Chengdu for Qingcheng Mountain
Recommended transport: High-speed train from Chengdu Xipu Station to Qingchengshan Station Duration: about 30 min Fare: 10 RMB Alternative: Self-drive Duration: about 1.5

hours Distance: 50 km 08:30–09:00 After arriving at Qingchengshan Station, head to the Tai'an Ancient Town parking lot
Take the scenic-area shuttle bus (16 RMB/person) or taxi (50 RMB/person) to the Rear Mountain entrance Buy Rear Mountain entrance tickets (20 RMB/person) Midday: Lunch in the Scenic Area 09:30–11:00 Lunch inside the scenic area Recommended: Eat at Youyicun (Youyi Village) or Tai'an Ancient Town Try local specialties such as smoked pork ribs and potatoes Budget: About 30–50 RMB per person Afternoon: Qingcheng Rear Mountain Hike 11:00–18:00 Hiking route on Qingcheng Rear Mountain
Route: Wulong Valley → Youyi Village → Baiyun Cave → Cuiying Lake → Feiquan Valley → Tai'an Ancient Town Route difficulty: Recommended Route 2 (about 70% difficulty), includes boat ride, cable car, and hiking Main sights: Wulong Valley: Beautiful scenery, best starting point for the hike Youyi Village: Small mountain village with rest stops and snacks Baiyun Cave: Worth exploring Cuiying Lake: Boat ride available (3 RMB/person), very scenic Feiquan Valley: Flowing streams and natural landscapes Cable car option: Baiyun Cableway (45 RMB/person one way) to save energy Photography tips: The section from Wulong Valley to Youyi Village has the best views Cuiying Lake and Feiquan Valley are also great for photography 18:00–19:00 Return to the scenic area entrance
Take a sightseeing bus (10 RMB/person) to the visitor center, then shuttle bus to Qingchengshan Station Or take a taxi (20 RMB/person) directly to Qingchengshan Station Evening: Return to Chengdu 19:30–21:00 Take the high-speed train back to Chengdu Day 2: In-Depth Exploration & Photography Morning: Depart from Chengdu 07:00–08:00 Depart from Chengdu for Qingcheng Mountain Same transport options as Day 1 Midday: Lunch in the Scenic Area 08:30–09:30 Lunch in the scenic area Same recommendations as Day 1 Afternoon: In-Depth Hiking on Qingcheng Rear Mountain 09:30–17:00 Deeper exploration of Qingcheng Rear Mountain
Route choice: You can try Route 3 (100% difficulty). Use caution; suitable for hikers with some experience. Key sights: Wanfo Cave: A whole rock wall covered with Buddha statues, very impressive Qianfo Cave: Also worth visiting Santan Wuquan (Three Pools and Misty Springs): A must-see classic spot Photography tips: Afternoon light is better for shooting caves and pools 17:00–18:00 Return to the scenic area entrance
Same transport options as Day 1 Evening: Return to Chengdu 18:30–20:00 Take the high-speed train back to Chengdu Cost Breakdown (Total 1,200 RMB for Two) Transportation High-speed train: Chengdu–Qingchengshan round trip About 40 RMB/person In-park transport: Scenic-area shuttle bus/taxi About 30–50 RMB/person Return transport: From Qingchengshan back to Chengdu About 40–60 RMB/person Tickets Qingcheng Rear Mountain entrance: 20 RMB/person (40 RMB for two) Cable car: Baiyun Cableway 45 RMB/person one way (round trip for two: 180 RMB, if needed) Cuiying Lake boat: 3 RMB/person (6 RMB for two) Food Meals in the scenic area: About 30–50 RMB/person (60–100 RMB for two) Snacks & supplies: Water, energy snacks, etc. About 50 RMB/person (100 RMB for two) Accommodation (If Extending the Trip) In-park lodging: Zhuolan Villa or local guesthouses About 200–300 RMB/night Recommended: Lanxipan Guesthouse in Dujiangyan Offers pick-up/drop-off and has a pleasant environment Photography Gear Camera: 800–1,000 RMB for two if renting (about 200–300 RMB/person) Lenses: About 500–800 RMB for two (if renting) Other gear: Tripod, filters, memory cards, etc. About 200–300 RMB/person Practical Tips Best visiting time
June–September: 08:00–18:00 (Mon–Sun) January–May & October–December: 08:00–17:30 Hiking gear
Comfortable, non-slip hiking shoes (essential) Lightweight backpack Drinking water and energy snacks Sun protection (hat, sunscreen, etc.) Camera tripod (essential for photography) Photography suggestions
Early morning and late afternoon light is best for landscapes After rain, Qingcheng Mountain is often shrouded in mist—excellent for photos Cuiying Lake and Feiquan Valley are popular photography spots Notes & safety

Mountain trails are steep; be careful and consider bringing trekking poles Toilets are scarce on the mountain; use water sparingly Mountain temperatures are lower than in the city; bring a jacket Check the weather forecast in advance and avoid heavy rain Keep your phone charged and communication available Wish you a wonderful trip and beautiful photos of Qingcheng Mountain!
</answer>

F.2 Case Study Following ArenaRL

Case Trajectory in Open-Travel Benchmark Following ArenaRL (translated into English)

Query: A two-day weekend hiking trip from Chengdu to Qingcheng Back Mountain for two people, with a focus on natural scenery and photography, and a budget of 600 RMB per person.

Step 1

<think>
The user wants to learn about a hiking itinerary from Chengdu to Qingcheng Back Mountain, suitable for two people, with a preference for natural scenery and photography, and a budget of 600 RMB per person. I need to first obtain basic information about Qingcheng Back Mountain, and then plan the travel route.
</think>
<tool_call>
{"name": "universal_search", "arguments": {"query": "Hiking Route from Chengdu to the Qingcheng Back Mountain, Natural Scenery, Photography"}}
</tool_call>
<tool_response>
...
</tool_response>

Step 2

<think>
I have already obtained the basic information and transportation options for hiking from Chengdu to Qingcheng Back Mountain. Now I need to further look up more detailed information, including the exact location of Qingcheng Back Mountain and the locations of its main scenic spots, in order to plan a reasonable hiking route. First, I need to use the search_poi tool to find the location information of Qingcheng Back Mountain.
</think>
<tool_call>
{"name": "search_poi", "arguments": {"query": "Qingcheng Back Mountain Scenic Area"}}
</tool_call>
<tool_response>
...
</tool_response>

Step 3

<think>
I have already obtained the basic information and location of Qingcheng Back Mountain. Now I need to continue searching for more information about the locations of the main scenic spots inside Qingcheng Back Mountain, in order to plan a reasonable hiking route. According to previous search results, the main attractions of Qingcheng Back Mountain include Tai'an Ancient Town, Wulong Valley, Youyicun, Cuiying Lake, etc. I need to look up the exact location of each of these spots one by one.
</think>
<tool_call>
{"name": "search_poi", "arguments": {"query": "Tai'an Ancient Town at the Qingcheng Back Mountain"}}
</tool_call>
<tool_response>
...
</tool_response>

Step 4

<think>
I have already obtained the basic information of Qingcheng Back Mountain and the location of Tai'an Ancient Town. Now I need to continue searching for the location of Wulong Valley, because it is one of the most important scenic spots in Qingcheng Back Mountain.
</think>
<tool_call>
{"name": "search_poi", "arguments": {"query": "Wulong Valley at the Qingcheng Back Mountain"}}
</tool_call>
<tool_response>
...
</tool_response>

Step 5

<think>
I have already obtained the basic information of Qingcheng Back Mountain, as well as the location information of Tai'an Ancient Town and Wulong Valley. Now I need to continue searching for other main attractions, such as Youyicun, Cuiying Lake, etc., in order to plan a complete hiking route.
</think>
<tool_call>
{"name": "search_poi", "arguments": {"query": "Youyicun at the Qingcheng Back Mountain"}}
</tool_call>
<tool_response>
...
</tool_response>

Step 6

<think>
I have obtained the location information of the main attractions of Qingcheng Back Mountain, including Tai'an Ancient Town, Wulong Valley, and Youyicun. Now I need to continue looking up the location of Cuiying Lake, and then plan a complete hiking route.
</think>
<tool_call>
{"name": "search_poi", "arguments": {"query": "Cuiying Lake at the Qingcheng Back Mountain"}}
</tool_call>
<tool_response>
...
</tool_response>

Step 7

<think>
I have collected the location information of the main attractions of Qingcheng Back Mountain, including Tai'an Ancient Town, Wulong Valley, Youyicun, and Cuiying Lake. Now I need to continue searching for some other important attractions, such as Laifeng Pavilion and Water Curtain Cave, in order to plan a complete hiking route.
</think>
<tool_call>
{"name": "search_poi", "arguments": {"query": "Laifeng Pavilion at the Qingcheng Back Mountain"}}
</tool_call>
<tool_response>
...
</tool_response>

Step 8

<think>
I have collected the location information of the main scenic spots in Qingcheng Back Mountain, including Tai'an Ancient Town, Wulong Valley, Youyicun, Laifeng Pavilion, and Cuiying Lake. Now I need to plan a reasonable hiking route and provide detailed navigation information. Based on the information collected, I will plan a hiking route starting from Tai'an Ancient Town, passing through Wulong Valley, Youyicun, Laifeng Pavilion, and Cuiying Lake, and finally returning to Tai'an Ancient Town. In this way, the route can cover the main natural scenic spots of Qingcheng Back Mountain and is suitable for photography enthusiasts.
Next, I need to use the get_navigation tool to obtain the navigation information for this route.
</think>
<tool_call>
{"name": "get_navigation", "arguments": {"start_lon": 103.490619, "start_lat": 30.920645, "end_lon": 103.490619, "end_lat": 30.920645, "via_points": "103.473101,30.923365;103.473823,30.934175;103.476093,30.931739;103.483087,30.936879"}}
</tool_call>
<tool_response>
...
</tool_response>

Step 9

<answer>
Chengdu to Qingcheng Rear Mountain Hiking Guide (Weekend 2-Day Trip)
Trip Overview Qingcheng Rear Mountain is one of the best hiking destinations around Chengdu, famous for its natural scenery and tranquil environment, and is especially suitable for photography enthusiasts. Below is a detailed 2-day itinerary to help you fully experience the beauty of Qingcheng Rear Mountain.
Transportation Getting to Qingcheng Mountain High-speed train: Take a high-speed train from Chengdu Xipu Station to Qingchengshan Station (about 30 minutes, 10 RMB per person one way). Self-drive: About 2 hours from Chengdu. Navigate to "Qingchengshan Rear Mountain Scenic Area Parking Lot" (there are Parking Area A and Parking Area B). Transport within the Scenic Area After getting off at Qingchengshan Station, you'll find tourist shuttle buses to the Front and Rear Mountains on the left side outside the station. Rear Mountain costs 15 RMB per person one way (about 40 minutes). Inside the scenic area there are sightseeing shuttle buses (10 RMB per person), or you can walk. Tickets Adult entrance ticket: 20 RMB per person Student ticket: 10 RMB per person Sightseeing shuttle bus in the scenic area: 10 RMB per person (one way) Luggage storage: 5 RMB per use Accommodation Suggestions Inside the Scenic Area Zhuolan Shanzhuang (Zhuolan Mountain Villa): Located at the foot of the mountain, offers mountain views, good service, and pick-up/drop-off service. Qixin Shanyin Guesthouse: Located in Taian Ancient Town, with a beautiful environment and attentive service. Youyicun Yuchengyuan: Located inside the Youyicun scenic area, quiet and secluded surroundings. Outside the Scenic Area There are many guesthouses and farmhouse inns in Taian Ancient Town, generally 200–500 RMB per night. Booking in advance is recommended.
Food and Dining Dining in the Scenic Area There are restaurants and snack stalls in the scenic area, but prices are relatively high. It's recommended to eat in the ancient town, where you can find 2-person set meals for around 30 RMB (one meat dish, one vegetable dish, one soup). Local specialties: farmhouse dishes, wild mushrooms from Qingcheng Mountain, free-range chicken, corn, etc. Bringing Your Own Food There are water sources and small shops along the hiking route, so you can bring some snacks and water. It's not recommended to carry too much food—keep your pack as light as possible. Recommended Hiking Route Route Overview Taian Ancient Town → Wulong Valley → Youyicun (Another Village) →

Laifeng Pavilion → Cuiying Lake → Taian Ancient Town
Detailed Itinerary Day 1
Morning
Depart from Chengdu for Qingcheng Mountain (by high-speed train or by car). After arriving at Qingchengshan Station, take the tourist shuttle bus to the Rear Mountain scenic area (about 40 minutes). Buy tickets at the scenic area entrance (20 RMB per person). Start hiking from Taian Ancient Town, following Qingcheng Mountain Road. Midday
Arrive at Wulong Valley, which features waterfalls, mountain springs, and ancient plank roads with beautiful scenery. Have lunch and rest near Wulong Valley (you can bring your own food or eat at a restaurant in the scenic area). Afternoon
Continue hiking to Youyicun; the scenery along this stretch is very pleasant. Visit the Youyicun scenic spot and enjoy the mountain and water views. Head to Laifeng Pavilion, an excellent viewpoint for sightseeing and photography. On the way, enjoy the scenery of Cuiying Lake; the water is clear and the environment serene. Return to Taian Ancient Town in the evening and have dinner there. Day 2
Morning
Depart from Taian Ancient Town and hike along Qingcheng Mountain Road. Focus on exploring Wulong Valley, the most scenic section of the Rear Mountain, with many waterfalls and mountain springs. You can choose to take a bamboo raft in Wulong Valley for a different way of sightseeing. Midday
Have lunch and rest in Youyicun, and sample local farmhouse specialties. Afternoon
Continue hiking to Cuiying Lake; the water is crystal clear and the surroundings are tranquil. Stay at Cuiying Lake for photos and to enjoy the reflections of the mountains in the water. Return to Taian Ancient Town in the evening, ending the trip. Recommended Photography Spots Wulong Valley: The most beautiful section of the Rear Mountain, with waterfalls, springs, and ancient plank roads. Bamboo rafting is available. The scenery is serene and highly photogenic. Laifeng Pavilion: Excellent vantage point for panoramic shots of the surrounding mountains and waters. Cuiying Lake: Clear water and tranquil surroundings, ideal for shooting reflections. Youyicun: A rustic village integrated with natural scenery, combining cultural and natural elements. Suggested Hiking Gear Comfortable hiking shoes—mountain trails are rugged, so footwear comfort is very important. Bring trekking poles—some sections are steep and narrow; the left-hand ascent offers better photo spots and can be tricky. Pack light; avoid unnecessary items. Bring enough drinking water, but note that there are very few toilets in the scenic area, so don't drink excessively. Bring a camera and spare batteries—Qingcheng Rear Mountain's natural scenery is excellent for photography. Practical Tips Best time to start: Set out around 8:30–9:00 a.m. to avoid peak crowds. Descent timing: Keep an eye on time; going down usually takes longer than going up. Weather: Check the forecast; trails can be slippery in the rain, so prepare for traction and safety. Sun protection: UV can be strong in the mountains; use sunscreen and wear a hat. Warm clothing: Even in summer it can be cool on the mountain; bring a light jacket. Pack light: Minimize what you carry so you can move easily. Environmental awareness: Take your trash with you and protect the natural environment. Budget Reference (Per Person) Transportation: Round-trip high-speed train 20 RMB + tourist shuttle bus 30 RMB = 50 RMB Tickets: Scenic area ticket 20 RMB + sightseeing shuttle bus 10 RMB (round trip) = 40 RMB Accommodation: 200–500 RMB (depending on guesthouse or farmhouse inn) Food: 30 RMB/day (for two people) × 2 days = 120 RMB Gear: Hiking shoes, trekking poles, photography gear, etc. about 100 RMB Other: Water, snacks, small souvenirs, etc. about 50 RMB Total: Approximately 460–610 RMB per person
This budget is for reference only; actual costs will vary based on accommodation choices and dining preferences. If you stay outside the scenic area or purchase equipment in advance, your total expenses may be lower.
Wish you a pleasant trip and wonderful photos of Qingcheng Rear Mountain!

</answer>

```
你是一名深谙旅游行业、具有严谨逻辑与评测方法论的「旅行规划 LLM 代理综合评审员」。现需对同一用户 Query 下，LLM Agent A 与 Agent B 的推理路径（Path）和回答结果（Answer
）分别进行分维度量化评估，并最终给出综合得分与胜者。请严格遵循下列指标、打分规则与输出格式。
一、评估内容格式
————————————————
<USER_QUERY>
{用户原始提问}
</USER_QUERY>

<PATH_A>
{LLM Agent A 的完整推理路径}
</PATH_A>

<PATH_B>
{LLM Agent B 的完整推理路径}
</PATH_B>

<ANSWER_A>
{LLM Agent A 的完整回答}
</ANSWER_A>

<ANSWER_B>
{LLM Agent B 的完整回答}
</ANSWER_B>
————————————————
二、推理路径评测（Path Evaluation）
————————————————
【评估维度说明】
1. 推理广度（Breadth）：是否全面覆盖用户需求，同时避免出现冗余或重复步骤。
2. 需求匹配度（Relevance）：各步骤与用户核心需求契合程度。
3. 细节信息丰富度（Detail）：引用的事实、数据、时间点、费用、预约规则等细节是否充分、准确且有用。

【评分规则】
• 推理路径评测时要求只关注推理路径中的实际工具调用，不用关注推理内容对信息的深入分析。
• 每个维度 0-10 分；0 表示"完全缺失"，10 表示"极为出色"。
• 推理路径综合得分（Overall_P）＝三个维度均值后四舍五入取整。
————————————————
三、回答结果评测（Answer Evaluation）
————————————————
【评估维度说明】
1. 匹配度（Relevance）：完整响应所有子需求/限制？顺序与场景贴合？
2. 可行性（Feasibility）：安排逻辑自洽、切实可行，避免明显冲突？
3. 细节丰富度（Details）：时间表、票价、交通耗时、Tips 等信息是否丰富且实用？
4. 清晰度（Clarity）：结构清晰、排版友好、可读性高？

【评分规则】
• 回答结果评测时需参考对应推理路径中的参考知识。
• 每个维度 0-10 分；0 表示"完全缺失"，10 表示"极为出色"。
• 回答结果综合得分（Overall_A）＝四个维度均值后四舍五入取整。
————————————————
四、综合得分与胜负判定
————————————————
综合得分 combined_scores = 0.6 * Overall_P（路径总体分） + 0.4 * Overall_A（答案总体分），四舍五入保留 1 位小数。
若 Combined 相同，则胜负判定结果为 Tie。
————————————————
【输出格式（严格遵循，不要添加多余内容）】
{
 "analysis": {
  "path_A": "<80-120 字中文评述：指出 A 路径亮点与不足>",
  "path_B": "<80-120 字中文评述：指出 B 路径亮点与不足>",
  "answer_A": "<80-120 字中文评述：指出 A 答案亮点与不足>",
  "answer_B": "<80-120 字中文评述：指出 B 答案亮点与不足>"
  },
 "path_scores": {
  "Agent_A": {
   "breadth": <0-10>,
   "relevance": <0-10>,
   "detail": <0-10>,
   "overall_p": <0-10>
  },
  "Agent_B": {
   "breadth": <0-10>,
   "relevance": <0-10>,
   "detail": <0-10>,
   "overall_p": <0-10>
  }
 },
 "answer_scores": {
  "Agent_A": {
   "relevance": <0-10>,
   "feasibility": <0-10>,
   "details": <0-10>,
   "clarity": <0-10>,
   "overall_a": <0-10>
  },
  "Agent_B": {
   "relevance": <0-10>,
   "feasibility": <0-10>,
   "details": <0-10>,
   "clarity": <0-10>,
   "overall_a": <0-10>
  }
 },
 "combined_scores": {
  "Agent_A": <0-10>,
  "Agent_B": <0-10>
 },
 "winner": "<Agent_A | Agent_B | Tie>"
}
【重要要求】
• 先逐维度独立思考后再给分，确保公平客观。
• 所有评语仅基于提供的文本，不要引入外部信息。
• 所有中文评述需具体、可溯源（可引用原文片段或段落号）。
• 严格遵守 JSON 模板，以便后续程序解析。

【工具解释】
- search_poi工具用于在一个指定的城市内搜索兴趣点（POI）的地理空间信息。
- around_search工具通过设置圆心和半径，搜索圆形区域内的地点信息。
- universal_search工具用于执行通用的、开放知识搜索。
- get_navigation工具除了起始点、终点经纬度，还可以设置via_points途经点。因此针对多点路线导航，既可以通过多次调用不带via_points的get_navigation来完成规划，也可以通过调用单次
带via_points的get_navigation来完成规划。因此评估应关注整条路线每个点是否都被覆盖到，在都覆盖了的前提下，再看路线信息的完整性，路线的合理性
```

Figure 5: Prompt of Open-Travel task.

```
你是一名精通信息检索方法论、具备严谨逻辑思维与系统化评测能力的「深度研究 LLM 代理综合评审员」。现需对同一用户 Query 下，LLM Agent A 与 Agent B 的研究路径（Path，
指首次回复中呈现的【研究步骤】及后续各轮工具调用日志）和最终回答（Answer，指完成全部检索后最后一次向用户展示的内容）进行分维度量化评估，并最终给出综合得分与
胜者。请严格遵循下列指标、打分规则与输出格式。
一、评估内容格式
————————————————
<USER_QUERY>
{用户原始提问}
</USER_QUERY>

<PATH_A>
{LLM Agent A 的完整研究路径}
</PATH_A>

<PATH_B>
{LLM Agent B 的完整研究路径}
</PATH_B>

<ANSWER_A>
{LLM Agent A 的完整回答}
</ANSWER_A>

<ANSWER_B>
{LLM Agent B 的完整回答}
</ANSWER_B>
————————————————
二、研究路径评测（Path Evaluation）
————————————————
【评估维度说明】
1. 研究框架完整性（Framework）： 是否在首轮给出条理清晰、递进合理、覆盖全面的研究步骤。
2. 工具调用策略（Tool Usage）： search_web 等工具调用是否针对性强、查询多样且无冗余，调用顺序与研究步骤匹配度高。
3. 信息覆盖完整性（Coverage）： 所检索信息能否充分覆盖用户需求，是否为后续回答奠定扎实依据，且没有重复冗余行动步骤。

【评分规则】
• 仅评估研究过程设计与工具使用本身，不评价其对资料的解读结果。
• 每维度 0-10 分；0=“完全缺失”，10=“极为出色”。
• 研究路径综合得分（overall_p）= 三个维度均值，四舍五入取整。
————————————————
三、回答结果评测（Answer Evaluation）
————————————————
【评估维度说明】
1. 契合度（Relevance）： 是否完整、准确地回应了用户所有问题与限制条件。
2. 事实准确性（Accuracy）： 关键数据、定义、结论是否充分，无明显错误或自相矛盾。
3. 论证深度（Depth）： 是否进行深入分析、比较与推理，展示批判性思考与清晰逻辑链。
4. 表达清晰度（Clarity）： 结构排版、用词与逻辑是否清晰易读，可直接为用户所用。

【评分规则】
• 评估时须参考对应研究路径所呈现的已检索信息；不得依据外部记忆。
• 每维度 0-10 分；0=“完全缺失”，10=“极为出色”。
• 回答结果综合得分（overall_a）= 四个维度均值，四舍五入取整。
————————————————
四、综合得分与胜负判定
————————————————
combined_scores = 0.5 × overall_p + 0.5 × overall_a，四舍五入保留 1 位小数。
若两者 combined_scores 相同，则判定为 Tie。
————————————————

【输出格式（严格遵循，不要添加多余内容）】
{
"analysis": {
"path_A": "<80-120 字中文评述：指出 A 路径亮点与不足>",
"path_B": "<80-120 字中文评述：指出 B 路径亮点与不足>",
"answer_A": "<80-120 字中文评述：指出 A 答案亮点与不足>",
"answer_B": "<80-120 字中文评述：指出 B 答案亮点与不足>"
},
"path_scores": {
"Agent_A": {
"framework": <0-10>,
"tool_usage": <0-10>,
"coverage": <0-10>,
"overall_p": <0-10>
},
"Agent_B": {
"framework": <0-10>,
"tool_usage": <0-10>,
"coverage": <0-10>,
"overall_p": <0-10>
}
},
"answer_scores": {
"Agent_A": {
"relevance": <0-10>,
"accuracy": <0-10>,
"depth": <0-10>,
"clarity": <0-10>,
"overall_a": <0-10>
},
"Agent_B": {
"relevance": <0-10>,
"accuracy": <0-10>,
"depth": <0-10>,
"clarity": <0-10>,
"overall_a": <0-10>
}
},
"combined_scores": {
"Agent_A": <0-10>,
"Agent_B": <0-10>
},
"winner": "<Agent_A | Agent_B | Tie>"
}
【重要要求】
• 先逐维度独立思考后再给分，确保公平客观。
• 所有评语仅基于提供的文本，不得引入外部信息。
• 评语需具体、可溯源（可引用原文片段或段落号）。
• 严格遵守 JSON 模板，以便后续程序解析。
```

Figure 6: Prompt of Open-DeepResearch task.

```
你是一名精通写作评估方法论、思考严谨的「写作 LLM 综合评审员」。现需对同一用户写作请求下，LLM A 与 B 的思考过程（Path）和最终写作结果（Answer）进行分维度量化评估，并最终给出综合得分与胜者。请严格遵循下列指标、打分规则与输出格式。
一、评估对象输入格式（评估时仅参考所给文本）
————————————————
<USER_QUERY>
{用户原始写作需求}
</USER_QUERY>

<PATH_A>
{LLM A 的完整思考/推理路径，含要点拆解、资料查找、结构构思等}
</PATH_A>

<PATH_B>
{LLM B 的完整思考/推理路径，含要点拆解、资料查找、结构构思等}
</PATH_B>

<ANSWER_A>
{LLM A 的最终文章/文案成品}
</ANSWER_A>

<ANSWER_B>
{LLM B 的最终文章/文案成品}
</ANSWER_B>
————————————————
二、推理路径评测 Path Evaluation（仅看 PATH，不看 ANSWER）
【维度说明】

1. 理解与拆解 (Understanding)：是否全面抓住用户所有写作要点、受众、限制。
2. 逻辑严谨性 (Logic)：思考步骤是否井然、论据与结论衔接顺畅，无跳步或自相矛盾。
3. 丰富度与创造性 (Richness)：是否从主题、结构、素材等多角度提出多种可选思路或素材。

【评分规则】
• 每维度 0-10 分；0 = “完全缺失”，10 = “极为出色”。
• 路径综合得分 overall_p = 三维度均值，四舍五入取整。

三、写作结果评测 Answer Evaluation（结合对应 PATH 参考知识）

【维度说明】

1. 需求契合度 (Relevance)：是否完整覆盖用户列出的全部要点，并严格满足用户设定的所有限制。
2. 内容质量与说服力 (Content Quality)：主题深度、论点论据充分性、吸引力与原创性。
3. 语言与文采 (Language & Style)：专业性、亲和力、流畅度，遣词造句精准多样。
4. 结构与可读性 (Clarity)：逻辑清晰、层次分明、排版友好。

【评分规则】
• 每维度 0-10 分；0 = “完全缺失”，10 = “极为出色”。
• 答案综合得分 overall_a = 四维度均值，四舍五入取整。

四、综合得分与胜负判定
1.combined_score = 0.4 × overall_p + 0.6 × overall_a，四舍五入保留 1 位小数。
2.若两个LLM combined_score 相同，则判定为 Tie。

五、输出格式（严格遵循，勿增删字段或更换顺序）

{
"analysis": {
"path_A": "<80-120 字中文评述：指出 A 路径亮点与不足>",
"path_B": "<80-120 字中文评述：指出 B 路径亮点与不足>",
"answer_A": "<80-120 字中文评述：指出 A 答案亮点与不足>",
"answer_B": "<80-120 字中文评述：指出 B 答案亮点与不足>"
},
"path_scores": {
"LLM_A": {
"understanding": <0-10>,
"logic": <0-10>,
"richness": <0-10>,
"overall_p": <0-10>
},
"LLM_B": {
"understanding": <0-10>,
"logic": <0-10>,
"richness": <0-10>,
"overall_p": <0-10>
}
},
"answer_scores": {
"LLM_A": {
"relevance": <0-10>,
"content_quality": <0-10>,
"language_style": <0-10>,
"clarity": <0-10>,
"overall_a": <0-10>
},
"LLM_B": {
"relevance": <0-10>,
"content_quality": <0-10>,
"language_style": <0-10>,
"clarity": <0-10>,
"overall_a": <0-10>
}
},
"combined_scores": {
"LLM_A": <0-10>,
"LLM_B": <0-10>
},
"winner": "<LLM_A | LLM_B | Tie>"
}

六、重要要求

1. 先逐维度独立思考再给分，保持客观一致。
2. 只依据提供文本，不引入外部信息或个人偏好。
3. 若写作结果中反复出现重复内容，或包含以<think>字符开头的推理路径，则应大幅降低答案得分。
4. 评述需具体可溯源，可引用“第 X 段”或原句关键词。
5. 必须输出合法 JSON，确保后续程序可解析。
```

Figure 7: Chinese prompt of open-ended writing task.

You are a rigorously minded “Comprehensive LLM Writing Evaluator” versed in writing-assessment methodology. Your task is to perform a multi-dimensional quantitative evaluation of two large language models—LLM A and LLM B—based on their reasoning processes (“Path”) and their final written outputs (“Answer”) produced in response to the same user request. Strictly follow the metrics, scoring rules and output format specified below.
I. Input Format for the Items to Be Evaluated
(When evaluating, refer only to the text provided)
———————————
<USER_QUERY>
{The user’s original writing request}
</USER_QUERY>

<PATH_A>
{LLM A’s complete chain of thought / reasoning path, including key-point breakdown, information retrieval, structural planning, etc.}
</PATH_A>

<PATH_B>
{LLM B’s complete chain of thought / reasoning path, including key-point breakdown, information retrieval, structural planning, etc.}
</PATH_B>

<ANSWER_A>
{LLM A’s final article / copy}
</ANSWER_A>

<ANSWER_B>
{LLM B’s final article / copy}
</ANSWER_B>
———————————

II. Path Evaluation (assess PATH only; ignore ANSWER)

[Dimension Descriptions]
1. Comprehension & Deconstruction (Understanding): Does the model fully capture all user requirements, target audience, and constraints?
2. Logical Rigour (Logic): Are the reasoning steps orderly, with smooth linkage between arguments and conclusions, free of gaps or contradictions?
3. Richness & Creativity (Richness): Does the model propose multiple viewpoints, structures, or materials from diverse angles?

[Scoring Rules]
• Each dimension: 0-10 points (0 = “entirely missing”, 10 = “outstanding”).
• Overall path score (overall_p) = arithmetic mean of the three dimensions, rounded to the nearest integer.

III. Answer Evaluation (assess ANSWER with reference to its PATH)
[Dimension Descriptions]

1. Requirement Alignment (Relevance): Does the piece fully address every point in the user brief and respect all specified constraints?
2. Content Quality & Persuasiveness (Content_Quality): Depth of insight, sufficiency of arguments/evidence, engagement, originality.
3. Language & Style (Language_Style): Professional tone, accessibility, fluency, precision and variety of expression.
4. Clarity & Readability (Clarity): Clear logic, well-structured sections, reader-friendly formatting.

[Scoring Rules]
• Each dimension: 0-10 points (0 = “entirely missing”, 10 = “outstanding”).
• Overall answer score (overall_a) = arithmetic mean of the four dimensions, rounded to the nearest integer.

IV. Combined Score & Winner Determination
1. combined_score = 0.4 × overall_p + 0.6 × overall_a, rounded to one decimal place.
2. If both models obtain the same combined_score, declare “Tie”.

V. Output Format (strictly follow; do not add, remove or reorder fields)
{
"analysis": {
"path_A": "<80-120 Chinese characters: highlight strengths and weaknesses of A’s path>",
"path_B": "<80-120 Chinese characters: highlight strengths and weaknesses of B’s path>",
"answer_A": "<80-120 Chinese characters: highlight strengths and weaknesses of A’s answer>",
"answer_B": "<80-120 Chinese characters: highlight strengths and weaknesses of B’s answer>"
},
"path_scores": {
"LLM_A": {
"understanding": <0-10>,
"logic": <0-10>,
"richness": <0-10>,
"overall_p": <0-10>
},
"LLM_B": {
"understanding": <0-10>,
"logic": <0-10>,
"richness": <0-10>,
"overall_p": <0-10>
}
},
"answer_scores": {
"LLM_A": {
"relevance": <0-10>,
"content_quality": <0-10>,
"language_style": <0-10>,
"clarity": <0-10>,
"overall_a": <0-10>
},
"LLM_B": {
"relevance": <0-10>,
"content_quality": <0-10>,
"language_style": <0-10>,
"clarity": <0-10>,
"overall_a": <0-10>
}
},
"combined_scores": {
"LLM_A": <0-10>,
"LLM_B": <0-10>
},
"winner": "<LLM_A | LLM_B | Tie>"
}

VI. Critical Requirements

1. Evaluate each dimension independently before assigning scores; remain objective and consistent.
2. Base judgments solely on the text supplied—introduce no outside information or personal preference.
3. If the answer output repeatedly contains duplicate content or includes reasoning paths starting with the <think> character, the answer_scores should be severely penalized.
4. Analytic comments must be traceable and specific; you may cite “paragraph X” or key phrases from the source.
5. Output must be valid JSON so that downstream programs can parse it.

Figure 8: English prompt of open-ended writing task.

角色设定
你是一名“航班查询结果模拟专家”，能够根据用户给出的日期、出发城市与到达城市，生成覆盖全天主要时段的机票信息（6-14 条）。所有信息均为模拟数据，但必须符合以下“真实性规则”。

输入格式
用户将以 JSON 形式输入：
{
"date": "YYYY-MM-DD",
"from_city": "出发城市中文名",
"to_city": "到达城市中文名"
}

输出格式
• 以 JSON 数组形式返回，每一条为一段中文字符串；
• 每条字符串遵循：
"航班 {航司代码+航班号}，价格{票价}元，{起飞时刻}从{出发机场}出发，{到达时刻}到达{到达机场}，飞行时长{X小时Y分}"
• 举例：
"航班 CA1847，价格763.0元，09:05从首都国际机场出发，12:25到达浦东国际机场，飞行时长3小时20分"

真实性规则

航司与航班号
• 航司代码：两位大写英文字母（常见：CA/MU/CZ/HU/HO/3U/GF/EK/AF 等）；
• 航班号：3-4 位数字。
机场
• 国内：使用城市主要机场（可带“国际 / 白塔 / 天府 / 首都 / 虹桥 / 禄口”等）；
• 国际：如有跨国城市，可使用国际机场（例：Heathrow、Changi、Narita 等）。
时间
• 出发时间覆盖 05:00-23:00，各航班间隔合理；
• 到达时间 = 出发时间 + 合理飞行时长（国内 1-4 小时，国际 2-15 小时）。
价格
• 国内：200-1500 元波动；
• 国际：800-8000 元波动；
• 同一日期票价从低到高大致递增但可随机。
条数
• 返回 10-15 条航班信息；
• 建议按起飞时间顺序排列，便于用户阅读。
语气
• 仅返回机票数组；不添加任何解释、换行、符号或多余信息。
示例交互
用户输入：
{"date":"2025-07-25","from_city":"呼和浩特市","to_city":"成都市"}

模型输出：
[
"航班 8L9672，价格745.0元，11:00从白塔国际机场出发，13:35到达天府机场，飞行时长2小时35分",
"航班 CA8147，价格763.0元，09:05从白塔国际机场出发，12:00到达天府机场，飞行时长2小时55分",
...
"航班 CA8131，价格965.0元，16:30从白塔国际机场出发，19:15到达天府机场，飞行时长2小时45分"
]

Figure 9: Prompt of search flights tool.

请扮演“火车票查询结果模拟器”。

输入是一段 JSON，字段包括：
• date：查询日期（格式 yyyy-MM-dd）
• from_city / to_city：中文城市名
• from_city_adcode / to_city_adcode：行政区划代码
• from_lat、from_lon、to_lat、to_lon：两地经纬度
任务：基于输入信息，输出 10-15 条该日期“{from_city}→{to_city}”的直达列车信息，覆盖凌晨、上午、下午、傍晚、夜间等大部分时段。
输出格式要求：
• 类型：JSON 数组，每个元素为一条车次信息字符串。
• 字符串内容模板：
“直达车次 {TrainNo}，价格{Price}元，{DepTime}从{DepStation}出发，{ArrTime}到达{ArrStation}，全程约{Duration}。”
• 关键值规范：
TrainNo：在 G / D / Z / K / T / Y / C 等字母+数字中随机选取，避免重复；
Price：综合里程与车种随机生成，动车/高铁 150-600 元，普速 60-300 元，硬卧可 100-420 元（仅普速时可给三档价位），车票价格根据两地距离而定；
DepTime / ArrTime：24h 制，确保 ArrTime ≥ DepTime，合理计算 Duration（四舍五入到分钟）；
DepStation / ArrStation：
• 如果城市内存在多个常见客运站（如“郑州”“郑州东”“郑州西”等），随机挑选符合列车类型的站名；
• 北/南/东/西/站字样请符合真实火车站命名习惯；
• Duration：按实际时间差给出“X时Y分”。
逻辑与随机性：
• 按常见列车运行规律生成时刻表，不要出现荒诞时间（如 03:00-03:20 只跑 20 分钟的普速）。
• 避免完全均匀分布，可略集中在早高峰 (06-09)、午后 (12-15)、晚高峰 (17-21) 等。
其他：
• 不输出与需求无关的文字、解释或注释，仅返回符合格式的 JSON 数组。
• 所有结果仅为模拟数据，非真实票务信息。

Figure 10: Prompt of search train tickets tool.