arXiv:2606.00593v2 [cs.CL] 2 Jul 2026

SPADER: Step-wise Peer Advantage with Diversity-Aware Exploration Rewards for Multi-Answer Question Answering

Qiming Shi[1,*], Zhaolu Kang[2,*], Yunfan Zhou[1], Di Weng[3,†], Yingcai Wu[1]
[1]State Key Lab of CAD&CG, Zhejiang University
[2]School of Software and Microelectronics, Peking University
[3]School of Software Technology, Zhejiang University
[*]Equal contribution [†]Corresponding author

Abstract

Large language models are increasingly deployed as tool-augmented agents to acquire information beyond parametric knowledge. While recent work has improved long-horizon tool-use reasoning, most approaches focus on tasks with a single correct answer. In contrast, many real-world queries require discovering a comprehensive set of valid answers, a setting known as Multi-Answer QA. This setting raises two challenges: fine-grained credit assignment over long search trajectories and reward alignment for sustained exploration beyond easy high-frequency entities. We propose **SPADER**, a reinforcement learning framework for long-horizon tool use in Multi-Answer QA. SPADER includes Step-wise Peer Advantage (SPA), a critic-free step-level credit assignment mechanism that aligns parallel trajectories by decision step and estimates advantages from peer returns. It also includes a diversity-aware exploration reward that promotes long-tail entity discovery by upweighting rare findings and downweighting redundant ones. Experiments on QAMPARI, Mintaka, WebQSP, and QUEST show that SPADER generally improves recall and overall F1 over prompting-based agents, outcome-supervised RL methods, and recent step-level supervision approaches. Our code and model weights are available at https://github.com/KhanCold/spader.

1 Introduction

Large language models (LLMs) are increasingly deployed as agents that interact with external tools to acquire information beyond their parametric knowledge. Recent work has extended retrieval-augmented generation (Lewis et al., 2020) into long-horizon reasoning loops in which models iteratively issue search queries, inspect retrieved evidence, and decide when to terminate with a final answer (Yao et al., 2023; Shao et al., 2023; Trivedi et al., 2023). This paradigm transforms question answering from a single retrieval step into a sequential decision-making process, where an agent must decide at each step whether to issue further queries or terminate with a final answer.

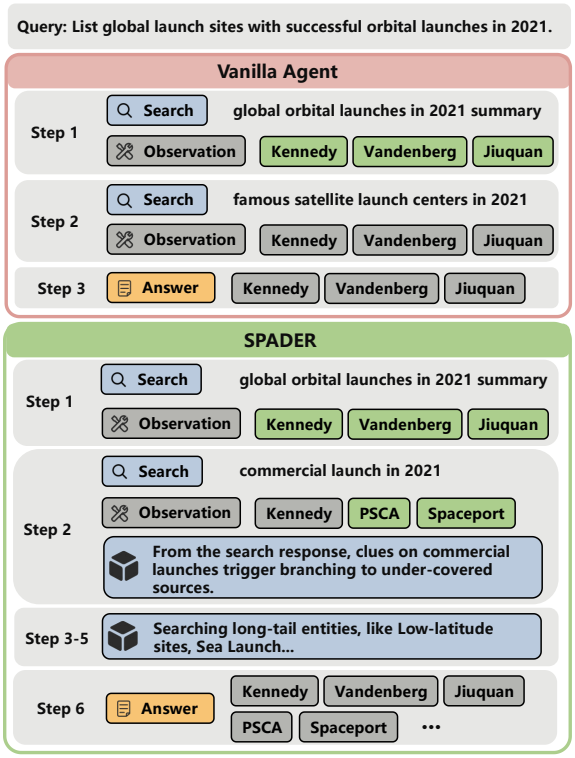

Figure 1: Vanilla Agent saturates on head entities and stops early, while SPADER keeps expanding toward long-tail launch sites under diversity-aware incentives.

Most existing work focuses on tasks that require only one correct answer. In contrast, many real-world information needs are coverage-oriented, requiring systems to discover a comprehensive set of valid answers. This setting, commonly referred to as Multi-Answer QA, shifts the central challenge from identifying a single correct answer to discovering a comprehensive set of valid answers. For example, given the query *"List global launch sites with successful orbital launches in 2021"* (Figure 1), an agent should not stop at prominent en-

tities like Kennedy Space Center. Instead, it must sustain exploration to uncover valid, long-tail answers like PSCA and Kourou.

To achieve such sustained exploration, agents must learn to make multi-step decisions, including when to reformulate queries, which evidence to incorporate, and when to terminate (Yao et al., 2023; Trivedi et al., 2023). Accordingly, recent work has increasingly framed tool-augmented QA as policy learning and optimized it with reinforcement learning (RL) (Li et al., 2025a; Jin et al., 2025). However, in Multi-Answer QA, existing RL approaches for tool-augmented reasoning still face two key limitations. First, long-horizon search trajectories make credit assignment difficult. As agents execute a mixture of productive, redundant, and irrelevant searches over extended steps, identifying which step genuinely expands the answer coverage is challenging. Actor-critic methods, such as PPO (Schulman et al., 2017), depend on value networks whose estimation error tends to compound over long horizons, making credit assignment unreliable and training costly. Critic-free methods, such as GRPO (Shao et al., 2024), improve stability, but their advantage estimation remains trajectory-level and thus provides only coarse credit assignment signal. Approaches that provide step-level feedback (Li et al., 2025b; Zheng et al., 2025a) often depend on external evaluators or process annotations, which limit scalability.

Second, commonly used reward formulations are poorly aligned with the objective of answer coverage. For instance, F1-based rewards credit every matched entity equally, providing no additional incentive for retrieving difficult, long-tail answers. Consequently, agents lack the motivation to sustain exploration for long-tail entities, resulting in premature halting after retrieving only prominent answers.

To address these challenges, we introduce **SPADER** (**S**tep-wise **P**eer **A**dvantage with **D**iversity-**A**ware **E**xploration **R**eward), an RL framework for long-horizon tool use in Multi-Answer QA. SPADER combines two complementary ideas. First, Step-wise Peer Advantage (SPA) provides a critic-free mechanism for fine-grained credit assignment. Instead of evaluating trajectories only at the sequence level, SPA aligns parallel trajectories by decision step and estimates advantages using the empirical future-return distribution of peer trajectories at the same step. This enables step-level policy optimization without requiring value networks or external evaluators. Second, a diversity-aware exploration reward explicitly encourages long-tail entity discovery. Beyond a base reward for retrieving valid entities, the reward for each entity is scaled inversely with its retrieval frequency across the trajectory group, encouraging exploration of new regions of the knowledge space while reducing redundant discoveries.

We evaluate SPADER on four Multi-Answer QA benchmarks: QAMPARI, Mintaka, WebQSP, and QUEST. Experimental results show that SPADER consistently improves recall and overall F1 compared with prompting-based agents, outcome-supervised RL approaches, and recent step-level supervision methods. These results highlight the importance of combining fine-grained credit assignment with diversity-aware exploration incentives when training long-horizon tool-use agents.

We summarize our contributions as follows:

- We propose Step-wise Peer Advantage (SPA), a critic-free step-level credit assignment mechanism that aligns parallel trajectories by decision step and derives advantages from peer trajectory return distributions.
- We introduce a diversity-aware exploration reward that encourages long-tail entity discovery by dynamically rewarding rare discoveries while reducing incentives for redundant high-frequency entities.
- Extensive experiments on four Multi-Answer QA benchmarks demonstrate that SPADER improves answer coverage and overall QA performance over strong prompting, outcome-supervised RL, and step-supervised baselines.

2 Related Work

2.1 Tool-Augmented QA Agents

Early open-domain QA agents largely rely on single-step retrieval pipelines (e.g., RAG), which retrieve a fixed context before generation (Guu et al., 2020; Lewis et al., 2020; Karpukhin et al., 2020; Xiong et al., 2021; Izacard and Grave, 2021; Izacard et al., 2023). These pipelines are effective for short factual queries but are less robust for iterative evidence gathering and multi-hop reasoning (Yang et al., 2018; Trivedi et al., 2022). Recent work therefore studies tool-augmented LLM agents that interleave reasoning and retrieval across multiple steps, including ReAct, Iter-RetGen, IR-CoT, and Search-o1 (Yao et al., 2023; Shao et al., 2023; Trivedi et al., 2023; Li et al., 2025a). These

agentic pipelines generally improve performance on knowledge-intensive and multi-hop QA benchmarks (Yang et al., 2018; Trivedi et al., 2022; Izacard et al., 2023).

2.2 Multi-Answer Question Answering

While most QA research targets single-answer scenarios, Multi-Answer QA (Amouyal et al., 2023; Malaviya et al., 2023; Sen et al., 2022; Yih et al., 2016) requires models to discover a comprehensive set of valid answers. Prior works attempt to improve answer coverage through various pipeline enhancements, including query decomposition (Min et al., 2020; Perez et al., 2020; Khot et al., 2023), iterative query expansion (Khattab et al., 2021), and diverse passage reading (Qi et al., 2021; Asai et al., 2022). However, these methods largely rely on heuristic combinations of static retrievers and readers, lacking dynamic mechanisms to systematically explore the long-tail search space.

Recent RL-driven reasoning agents further extend tool-augmented QA by learning autonomous policies for retrieval and reasoning. Systems such as DeepRAG (Guan et al., 2026), R1-Searcher (Song et al., 2025), Search-r1 (Jin et al., 2025), and DeepResearcher (Zheng et al., 2025b) demonstrate strong performance on complex QA. However, standard reward formulations in these systems typically assign equal value to every correct entity found regardless of their retrieval difficulty. As a result, existing agents may favor easy-to-retrieve entities and under-explore long-tail answers that remain discoverable.

2.3 Fine-Grained Credit Assignment in RL

Effective credit assignment in long-horizon reasoning requires attributing sparse rewards to intermediate steps (Sutton and Barto, 2018; Arjona-Medina et al., 2019). Classical actor-critic training can be unstable or expensive in this regime due to value estimation over long trajectories (Schulman et al., 2017; Mnih et al., 2016; Haarnoja et al., 2018). Step-level supervision with PRMs and process-guided retrieval improves feedback granularity (Lightman et al., 2024; Uesato et al., 2022; Li et al., 2025b; Zheng et al., 2025a), but often requires costly annotation pipelines or strong external teachers. In a complementary direction, critic-free methods such as GRPO estimate relative advantages from grouped trajectories (Shao et al., 2024; Ahmadian et al., 2024), improving optimization efficiency but still aggregating credit at the trajectory level. While recent works (e.g., GiGPO, SALT) achieve step-level credit assignment via exact state matching (Feng et al., 2025; Li et al., 2026), this strict requirement limits their applicability in Multi-Answer QA due to the inherently noisy and highly variable search observations. In contrast, our SPA avoids brittle state matching by directly aligning trajectories by decision step.

3 Task Formulation

We formulate the LLM-based long-horizon tool-use for Multi-Answer QA as a Markov Decision Process (MDP). Given a query q and its ground-truth entity set $\mathcal{E}_{\text{GT}}$, the agent interacts with the environment via multi-step search calls to output a complete and accurate entity list $\mathcal{E}_{\text{final}}$.

At step t, the state $s_t{=}[q, a_1, o_1, \ldots, a_{t-1}, o_{t-1}]$ concatenates q and the interaction history, where a_{t-1} is the previous action and o_{t-1} is the textual observation returned by the search execution. Following policy $\pi_\theta(a_t|s_t)$, the agent selects an action from $\mathcal{A} = \{a_{\text{search}}, a_{\text{answer}}\}$:

- **Search Action** (a_{search}): The agent generates a query to invoke an external search engine. The environment returns relevant document snippets as observation o_t, transitioning the state to s_{t+1} to expand knowledge coverage without triggering the final answer generation.
- **Terminal Action** (a_{answer}): Based on the accumulated state s_t, the agent outputs the final predicted entity set $\mathcal{E}_{\text{final}}$. This acts as the absorbing state, immediately terminating the current trajectory.

Thus, a full trajectory is strictly denoted as an alternating sequence $\tau = (s_1, a_1, o_1, \ldots, s_L, a_L)$, where L is the terminal step.

4 SPADER

To address the dual challenges of credit assignment and coverage-oriented exploration in long-horizon Multi-Answer QA, we propose SPADER (**S**tep-wise **P**eer **A**dvantage with **D**iversity-Aware **E**xploration **R**eward). Building on GRPO's group-based mechanism, SPADER shifts the advantage estimation from the standard trajectory level to a fine-grained, step-wise level.

Specifically, SPADER consists of two core components: a Step-wise Peer Advantage (SPA) for fine-grained credit assignment, and a Diversity-Aware Exploration Reward to prevent premature convergence to head entities.

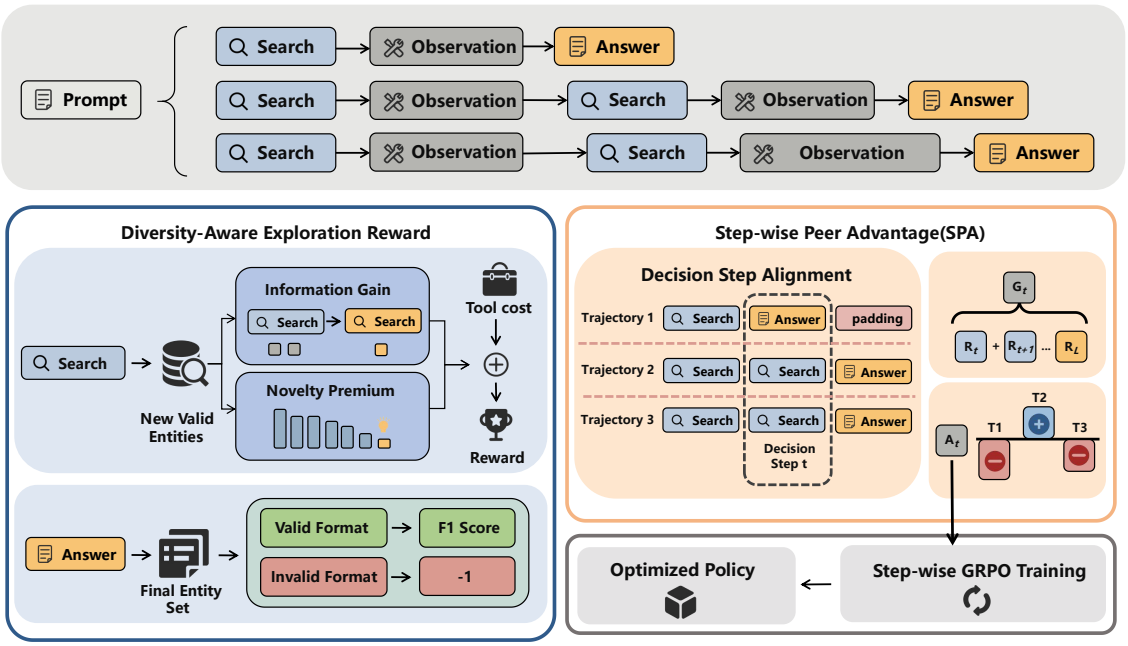

Figure 2: Overview of the SPADER framework. After sampling parallel trajectories, the framework evaluates each Search and Answer action using a Diversity-Aware Exploration Reward. Subsequently, Step-wise Peer Advantage (SPA) aligns these trajectories by decision steps to compute future returns (G_t) and step-wise advantages, which jointly optimize the policy through step-wise GRPO.

4.1 Step-wise Peer Advantage (SPA)

Decision Step Alignment. To establish a baseline for step-level credit assignment without a parameterized value network, we align parallel trajectories by their *decision step* t. Formally, a decision step t corresponds to the agent emitting an action $a_t \in \{a_{\text{search}}, a_{\text{answer}}\}$ given the accumulated state s_t. Since trajectories sampled under the same prompt consume identical interaction turns at step t, aligning them establishes a prompt- and interaction-budget-conditioned baseline. This evaluates whether an action yields a higher future return than the group expectation under the exact same interaction budget, without assuming semantic or state equivalence across trajectories.

To handle varying lengths, we pad prematurely terminated trajectories to the maximum length of the group $L_{\max}^{(\text{Group})}$. They participate in the group baseline with a future return of 0, while a Validity Mask $M_t^{(i)}$ blocks invalid gradient backpropagation after termination:

$$M_t^{(i)} = \begin{cases} 1, & t \le L_i \text{ (Active)} \\ 0, & t > L_i \text{ (Terminated)} \end{cases} \tag{1}$$

Advantage Estimation. To achieve fine-grained step-level credit assignment, SPA leverages the future cumulative return distribution of the parallel trajectory group at the equivalent decision step as a dynamic baseline.

First, for the action taken by trajectory i at step t, its actual future cumulative discounted return is calculated as $G_t^{(i)} = \sum_{k=0}^{L_i - t} \gamma^k r_{t+k}^{(i)}$, where γ is the discount factor, and the summation continues until the actual termination step L_i.

Subsequently, we consider the G trajectories sampled under the same prompt as a comparison group. At decision step t, the group baseline μ_t represents the expected empirical future return across the entire group: $\mu_t = \frac{1}{G} \sum_{j=1}^{G} \left(G_t^{(j)} \cdot M_t^{(j)} \right)$.

By comparing the individual return against the group empirical distribution and applying standardization, we obtain the step-wise relative advantage $\hat{A}_t^{(i)}$ for trajectory i at step t:

$$\hat{A}_t^{(i)} = \frac{G_t^{(i)} - \mu_t}{\sigma_t} \cdot M_t^{(i)} \tag{2}$$

where σ_t is the standard deviation of the group returns at step t.

While standard Monte Carlo returns can exhibit high variance due to the compounding noise of future actions, SPA is designed to alleviate this issue through its step-aligned empirical baseline. By subtracting the mean group return μ_t at the corresponding decision node, this formulation accounts

for the shared expectation of future trajectory outcomes.

Such relative comparison yields a more localized estimate of the current action's utility, assigning positive advantages to steps that empirically outperform their peers. Consequently, SPA facilitates fine-grained credit assignment for long-horizon search without necessitating parameterized value networks or external evaluators.

Training Objective. Our training objective adapts the GRPO framework (Shao et al., 2024) to a step-wise setting. By replacing the sequence-level advantage with our Step-wise Peer Advantage $\hat{A}_t^{(i)}$, we maximize the following objective:

$$\mathcal{J}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\sum_{t=1}^{L_i}\Bigg[\min\Big(\rho_t^{(i)}(\theta)\hat{A}_t^{(i)}, \text{clip}\left(\rho_t^{(i)}(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t^{(i)}\Big) - \beta_{KL}\mathbb{D}_{KL}\Bigg] \quad (3)$$

where $\rho_t^{(i)}(\theta) = \frac{P_\theta(y_t^{(i)}|x_t^{(i)})}{P_{\theta_{old}}(y_t^{(i)}|x_t^{(i)})}$ represents the probability ratio between the current and old policies. $\mathbb{D}_{KL}$ denotes the KL divergence between the active and reference models to prevent excessive policy deviation, with β_{KL} controlling the penalty strength.

4.2 Diversity-Aware Exploration Reward

To incentivize the discovery of long-tail entities and prevent premature convergence, we structure the multi-step reward into two distinct phases: a Dual-Axis exploration reward during the active search phase, and a global evaluation reward for the final generation phase.

4.2.1 Dual-Axis Reward for Exploration

For each search action ($a_t = a_{\text{search}}$), we evaluate its exploration utility by decoupling the step-level reward into two orthogonal dimensions: horizontal information gain and vertical novelty.

For a newly discovered valid entity $e \in \mathcal{E}_{\text{new},t}^{(i)}$ acquired at step t by trajectory i, its reward $R_{\text{ent}}(e)$ is a weighted sum of two utilities:

$$R_{\text{ent}}(e) = \alpha \cdot U_{\text{gain}}(e) + \beta \cdot U_{\text{novelty}}(e) \quad (4)$$

where $\alpha, \beta > 0$ are balancing hyperparameters.

Specifically, the two utility components are defined as follows:

- **Horizontal Information Gain (U_{gain}):** Represents the fundamental utility of retrieving a valid new entity, independent of peer behavior. We define this as a constant: $U_{\text{gain}}(e) = 1$.
- **Vertical Novelty Premium (U_{novelty}):** A reward dilution mechanism to penalize redundant discoveries across the parallel trajectory group $\mathcal{T}$. Let $N(e, \mathcal{T}) = \sum_{j=1}^{G} \mathbf{1}(e \in \tau_j)$ be the number of trajectories in the group retrieving e. The novelty utility is inversely proportional to its group-level frequency: $U_{\text{novelty}}(e) = 1/N(e, \mathcal{T})$.

This formulation establishes a dynamic incentive structure directly aligned with the coverage-oriented objective of Multi-Answer QA. Highly accessible "head" entities, which are easily discovered by the majority of parallel trajectories ($N \approx G$), face severe novelty dilution. Their marginal utility is consequently dominated merely by the base gain α. Conversely, unique long-tail discoveries ($N = 1$) yield the maximum reward $\alpha + \beta$.

By explicitly tying the reward scale to an entity's group-level frequency, this mechanism naturally prevents the agent from greedily settling for high-frequency targets and terminating early. Instead, the policy retains a continuous incentive to offset the search costs and sustain exploration to uncover the comprehensive, long-tail answer distribution.

The overall step-level reward for trajectory i, upon executing the search action $a_t^{(i)}$, is aggregated as:

$$r_t^{(i)} = \frac{1}{|\mathcal{E}_{\text{GT}}|}\Bigg[\sum_{e \in \mathcal{E}_{\text{new},t}^{(i)}} R_{\text{ent}}(e) - \text{Cost}(a_t^{(i)})\Bigg] \quad (5)$$

where $\text{Cost}(a_t^{(i)})$ imposes a fixed tool cost to discourage redundant or uninformative queries. Normalizing the total step reward by the ground-truth size $|\mathcal{E}_{\text{GT}}|$ explicitly bounds the absolute reward scale, ensuring stable gradient updates during the RL process.

4.2.2 Terminal Settlement Reward

Upon executing the terminal action ($a_t^{(i)} = a_{\text{answer}}$), the agent ceases search to output the final entity set $\mathcal{E}_{\text{final}}^{(i)}$. We define the terminal reward $r_{L_i}^{(i)}$ as:

$$r_{L_i}^{(i)} = \begin{cases} -1, & \text{Invalid Format} \\ \text{F1}(\mathcal{E}_{\text{final}}^{(i)}, \mathcal{E}_{\text{GT}}), & \text{Otherwise} \end{cases} \quad (6)$$

Methods	QAMPARI			Mintaka			WebQSP			QUEST		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Llama3.1-8B												
RAG	0.336	0.118	0.157	0.330	0.425	0.355	0.289	0.304	0.268	0.129	0.066	0.075
ReAct	0.388	0.202	0.241	0.409	0.481	0.424	0.323	0.313	0.286	0.151	0.070	0.082
PPO	0.353	0.253	0.268	0.610	0.595	0.600	0.325	0.368	0.329	0.192	0.108	0.126
GRPO	0.233	0.180	0.187	0.540	0.527	0.527	0.349	0.374	0.352	0.104	0.052	0.059
R3-RAG	0.386	0.290	0.303	0.617	0.630	0.607	0.452	0.413	0.403	0.169	0.117	0.121
StepSearch	0.418	0.258	0.304	0.639	0.612	0.618	0.499	0.387	0.403	0.181	0.104	0.124
SPADER (QAMPARI-only)	0.420	**0.411**	**0.394**	0.538	0.594	0.558	0.427	0.392	0.378	**0.216**	0.112	0.128
SPADER	**0.458**	0.340	0.348	**0.642**	**0.644**	**0.625**	**0.503**	**0.453**	**0.438**	0.194	**0.126**	**0.131**
Qwen3-8B												
RAG	0.181	0.119	0.128	0.297	0.448	0.325	0.314	0.284	0.273	0.154	0.065	0.078
ReAct	0.419	0.173	0.218	0.384	0.466	0.404	0.344	0.351	0.318	0.158	0.077	0.090
PPO	0.410	0.256	0.279	0.573	0.546	0.550	0.380	0.340	0.336	0.176	0.100	0.109
GRPO	0.423	0.182	0.226	0.528	0.497	0.504	0.339	0.290	0.290	0.167	0.092	0.104
R3-RAG	0.417	0.254	0.276	0.587	0.559	0.563	0.385	0.343	0.338	0.185	0.105	0.114
StepSearch	0.429	0.304	0.317	0.585	0.562	0.565	0.377	0.340	0.333	0.194	**0.119**	0.126
SPADER (QAMPARI-only)	**0.505**	**0.336**	**0.378**	0.511	0.536	0.513	0.351	0.328	0.321	0.211	0.103	0.122
SPADER	0.474	0.312	0.343	**0.625**	**0.626**	**0.611**	**0.423**	**0.409**	**0.388**	**0.233**	0.109	**0.127**

Table 1: Performance comparison of our proposed approach against various baselines across four Multi-Answer QA datasets. We report Precision (P), Recall (R), and F1 scores for both Llama3.1-8B and Qwen3-8B backbones. Baseline methods are grouped into Prompting-based, Outcome-Supervised RL, and Step-level Supervision Evaluators.

The terminal reward $r_{L_i}^{(i)}$ enforces structural compliance and closes the task loop. By providing a fixed evaluation of the final output, it serves as a necessary exit signal; when the expected novelty reward no longer outweighs the cumulative transition costs, this global constraint incentivizes the agent to cease exploration and deliver the final answer.

5 Experiments

5.1 Datasets and Evaluation Metrics

Our empirical benchmarks span four established Multi-Answer QA datasets: (1) QAMPARI (Amouyal et al., 2023), (2) Mintaka (Sen et al., 2022), (3) WebQSP (Yih et al., 2016), and (4) QUEST (Malaviya et al., 2023). More details about the datasets are provided in Appendix B.

For multi-answer evaluation, we compute instance-level Precision, Recall, and F1 based on the overlap between the set of unique predicted answers and the ground-truth answer set under exact match (EM) constraints. We report macro-averaged Precision, Recall, and F1 over all instances.

5.2 Baselines

To comprehensively evaluate our approach, we compare against several competitive baselines grouped into three established paradigms: (1) **Prompting-based Methods**: Approaches utilizing the frozen base model without policy optimization, including **RAG** (Lewis et al., 2020) and **ReAct** (Yao et al., 2023); (2) **Outcome-Supervised RL**: Frameworks optimizing sparse terminal rewards based on final correctness, comprising **PPO** (Schulman et al., 2017) and **GRPO** (Shao et al., 2024); and (3) **Step-level Supervision Evaluators**: Methods that explicitly score intermediate retrieval and reasoning steps, **R3-RAG** (Li et al., 2025b), and **StepSearch** (Zheng et al., 2025a).

5.3 Implementation Details

We instantiate our proposed algorithm across two different models, utilizing Llama3.1-8B-Instruct (Grattafiori et al., 2024) and Qwen3-8B (Yang et al., 2025) as the backbone language models. For the search tool, we deploy a batched two-stage retrieval pipeline over the Wikipedia 2021 dump (aligning with the QAMPARI setup). Upon receiving a batch

of queries, the system processes each query individually by fetching the top-50 candidates via a local BM25 retriever (Robertson and Zaragoza, 2009), followed by re-ranking with BGE-Reranker-v2-M3 (Chen et al., 2024) to select the top-5 passages.

During the training phase, the models are trained on 12,000 instances sampled from the train splits: 6,000 from QAMPARI and 2,000 each from Mintaka, WebQSP, and QUEST. To compute step-level rewards, after each rollout we normalize tool-response and ground-truth entities and then apply exact matching at each step; newly discovered entities at step t are matched entities not seen in earlier steps. The prompt template and further details are provided in Appendix C.

5.4 Main Results

Table 1 presents the evaluation results across the four Multi-Answer QA benchmarks. SPADER performs better than the compared baselines on both Llama3.1-8B and Qwen3-8B, achieving the best overall F1 scores.

Comparison with Prompting and Outcome-Supervised RL. Prompting methods (RAG, ReAct) and outcome-supervised RL (PPO, GRPO) show lower performance, particularly in Recall. For instance, on the QAMPARI dataset (Llama3.1-8B), SPADER achieves an F1 of 0.348, yielding relative improvements of 29.9% and 86.1% over PPO and GRPO, respectively. These gaps suggest that step-wise credit assignment can be more effective for long-horizon search than relying only on terminal rewards or sequence-level returns.

Comparison with Step-Level Supervision. When compared to methods utilizing step-level supervision (R3-RAG, StepSearch), SPADER also yields consistent improvements. This suggests that dynamically deriving advantages from intra-group empirical alignments (SPA) may provide a more calibrated learning signal than using external evaluators or static heuristics.

Model and Dataset Generalization. SPADER consistently achieves the highest F1 scores among all baselines on both Llama3.1-8B and Qwen3-8B across all datasets, showing robust, model-agnostic generalization across different architectures. Furthermore, **SPADER (QAMPARI-only)**, trained on 12,000 instances, remains competitive on unseen domains relative to prompting baselines.

6 Analysis

6.1 Ablation study

Table 2 evaluates the contribution of each SPADER component on Qwen3-8B. Overall, removing any component degrades performance, indicating that the gains are not driven by a single design choice.

The w/o SPA setting causes a stable decline across datasets, supporting the role of Step-wise Peer Advantage in improving long-horizon action quality. The w/o Novelty Premium setting leads to broad performance degradation, showing that the Vertical Novelty Premium is necessary for avoiding early collapse to frequent entities.

At the same time, the w/o Info Gain setting remains weak despite retaining the novelty term, with especially large drops on Mintaka and QAMPARI. This pattern suggests that novelty alone can overbias the policy toward long-tail entities when Horizontal Information Gain is removed. Taken together, these results indicate that the Dual-Axis Reward for Exploration must be balanced, and the full SPADER objective is needed to achieve strong exploration quality without sacrificing final answer accuracy. Full Precision/Recall/F1 ablation results are provided in Appendix Table 4.

Methods	QAMPARI	Mintaka	WebQSP	QUEST
SPADER	**0.343**	**0.611**	**0.388**	**0.127**
w/o SPA	0.285	0.569	0.336	0.117
w/o NP	0.239	0.373	0.256	0.083
w/o IG	0.266	0.364	0.359	0.094

Table 2: Ablation study on Qwen3-8B. We report F1 on four datasets. "w/o" denotes removing one component from SPADER. NP and IG denote novelty premium and information gain, respectively.

6.2 Hyperparameter Sensitivity Analysis

We analyze the sensitivity of SPADER to the novelty weight β in the dual-axis exploration reward. All settings are fixed to the main configuration, and we sweep $\beta \in \{0, 1, 4, 8\}$ on Qwen3-8B with $\alpha = 1$. We do not additionally sweep α. In our reward design, α is a global scaling factor on the base gain term, while β directly controls how strongly the policy prefers rare entities.

As shown in Table 3, removing novelty reward ($\beta = 0$) substantially hurts F1 on all datasets. Increasing β from 0 to 4 consistently improves performance, with the best results achieved at the main setting $\beta = 4$. When β is further increased to 8,

performance plateaus and slightly declines, suggesting that overly strong novelty pressure can bias exploration toward rare but less useful entities.

β	QAMPARI	Mintaka	WebQSP	QUEST
0	0.239	0.373	0.256	0.083
1	0.331	0.594	0.371	0.120
4	**0.343**	**0.611**	**0.388**	**0.127**
8	0.336	0.602	0.381	0.123

Table 3: Sensitivity of SPADER to the novelty weight β on Qwen3-8B. We report F1 on four datasets.

6.3 Search Efficiency Analysis

To examine whether the performance gains come with excessive exploration, we analyze the average search counts required to retrieve a single valid entity. The results on the Qwen3-8B are illustrated in Figure 3.

While SPADER does not exhibit the lowest search counts per entity, it maintains competitive search efficiency while improving overall performance metrics (as shown in Table 1). Methods with higher search counts exhibit different behaviors. GRPO requires the highest search count per valid entity but yields poor recall, suggesting it struggles with redundant search calls.

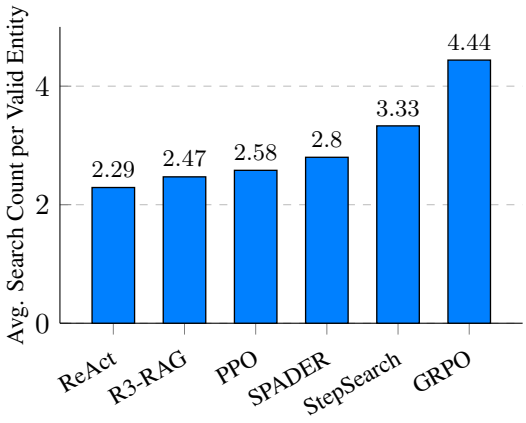

Figure 3: Search efficiency comparison on the Qwen3-8B backbone. The y-axis represents the average search calls needed to yield one valid entity.

In contrast, baselines with lower search counts per entity, including ReAct, R3-RAG, and PPO, show lower recall. This suggests that their seemingly high efficiency may partly come from earlier termination, where the agent captures head entities but explores less for long-tail ones.

6.4 Temporal Exploration Dynamics

To evaluate whether our approach effectively sustains exploration to uncover long-tail entities, we analyze the temporal discovery dynamics. Figure 4 tracks the cumulative number of distinct valid entities across interaction turns on the QAMPARI dataset using the Qwen3-8B backbone.

As Figure 4 shows, methods like GRPO and ReAct converge prematurely after the initial turn. While StepSearch extends its search into subsequent turns, SPADER maintains a higher cumulative yield across the entire trajectory. It captures more long-tail entities through a higher initial discovery and continued exploration in early turns, eventually plateauing as valid targets are exhausted.

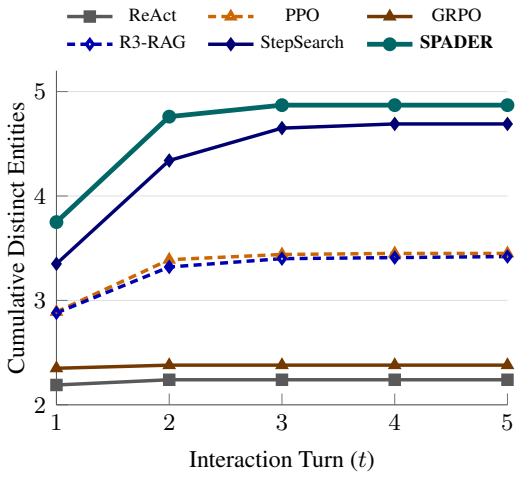

Figure 4: Cumulative distinct entities per interaction turn on QAMPARI.

6.5 Case Study

We provide four qualitative cases in the Appendix E to complement the quantitative results, covering comparative evidence against baselines, behavior changes observed in ablations, and one failure case.

7 Conclusion

We study long-horizon tool use for Multi-Answer QA as a RL problem where the primary challenge lies in coverage-oriented exploration. To address this, we propose SPADER, a framework combining Step-wise Peer Advantage (SPA) for critic-free step-level credit assignment with a diversity-aware exploration reward for long-tail entity discovery. Experiments across four benchmarks show that SPADER effectively mitigates premature termination, consistently outperforming existing prompting-based and RL-driven agents.

Limitations

Despite the effectiveness of SPADER, our study has three main limitations. First, the experiments are conducted in an offline Wikipedia-based retrieval setting, which may not fully reflect the complexities of live web environments, such as content noise, API latency, and evolving knowledge. Second, we evaluate our approach using 8B-parameter backbones on four QA benchmarks. Broader validation across larger model scales, multilingual contexts, and diverse tool-use tasks remains an important direction for future work. Third, our reliance on coarse exact-match entity detection may miss lexical variants. Future extensions could incorporate task-specific canonicalization or LLM-based verification for more robust reward signals.

Ethical Consideration

This work adheres to the ACL Ethics Policy. The benchmarks employed in our study are derived from publicly available and de-identified sources, ensuring no privacy risks or exposure of personally identifiable information. Regarding potential impact, although SPADER enhances answer coverage, the system may reflect intrinsic biases or factual inaccuracies present in the underlying language models and retrieved web corpora. We recommend exercising caution when deploying such agents in sensitive or high-stakes information-seeking domains.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, Bangkok, Thailand. Association for Computational Linguistics.

Samuel Amouyal, Tomer Wolfson, Ohad Rubin, Ori Yoran, Jonathan Herzig, and Jonathan Berant. 2023. QAMPARI: A benchmark for open-domain questions with many answers. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 97–110, Singapore. Association for Computational Linguistics.

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. 2019. Rudder: Return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Akari Asai, Matt Gardner, and Hannaneh Hajishirzi. 2022. Evidentiality-guided generation for knowledge-intensive NLP tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2226–2243, Seattle, United States. Association for Computational Linguistics.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. Group-in-group policy optimization for LLM agent training. In *Advances in Neural Information Processing Systems*, volume 38, pages 46375–46408. Curran Associates, Inc.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. 2026. DeepRAG: Thinking to retrieve step by step for large language models. In *The Fourteenth International Conference on Learning Representations*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard

Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Baleen: Robust multi-hop reasoning at scale via condensed retrieval. In *Advances in Neural Information Processing Systems*, volume 34, pages 27670–27682. Curran Associates, Inc.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Jiazheng Li, Yawei Wang, Qiaojing Yan, Yijun Tian, Zhichao Xu, Huan Song, Panpan Xu, and Lin Lee Cheong. 2026. SALT: Step-level advantage assignment for long-horizon agents via trajectory graph. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 4709–4725, Rabat, Morocco. Association for Computational Linguistics.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025a. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5420–5438, Suzhou, China. Association for Computational Linguistics.

Yuan Li, Qi Luo, Xiaonan Li, Bufan Li, Qinyuan Cheng, Bo Wang, Yining Zheng, Yuxin Wang, Zhangyue Yin, and Xipeng Qiu. 2025b. R3-RAG: Learning step-by-step reasoning and retrieval for LLMs via reinforcement learning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10491–10507, Suzhou, China. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.

Chaitanya Malaviya, Peter Shaw, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2023. QUEST: A retrieval dataset of entity-seeking queries with implicit set operations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14032–14047, Toronto, Canada. Association for Computational Linguistics.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937. PMLR.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8864–8880, Online. Association for Computational Linguistics.

Peng Qi, Haejun Lee, Tg Sido, and Christopher Manning. 2021. Answering open-domain questions of varying reasoning steps from text. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3599–3614, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Priyanka Sen, Alham Fikri Aji, and Amir Saffari. 2022. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1604–1619, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274, Singapore. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Yang Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, pages 1279–1297, New York, NY, USA. Association for Computing Machinery.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement Learning: An Introduction*, second edition. MIT Press.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.

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

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany. Association for Computational Linguistics.

Xuhui Zheng, Kang An, Ziliang Wang, Yuhang Wang, and Yichao Wu. 2025a. StepSearch: Igniting LLMs search ability via step-wise proximal policy optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21805–21830, Suzhou, China. Association for Computational Linguistics.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025b. DeepResearcher: Scaling deep research via reinforcement learning in real-world environments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 414–431, Suzhou, China. Association for Computational Linguistics.

A Prompt Template

To ensure the backbone Large Language Models properly format their multi-step reasoning and tool-use procedures, we use a structured system prompt. As illustrated in Figure 5, the prompt enforces a strict thought-action observation loop. The agent is required to first output its reasoning traces within the <think> tags, followed by either a retrieval action (<tool_call>) or a final answer generation (<answer>). Both outputs are strictly constrained to a JSON list format to facilitate robust parsing.

System Prompt Template

```
Answer the question. You must strictly follow this
loop for every step:
1. Always start with reasoning inside <think>...</think>.
2. Then, output either a search tool call OR the
   final answer.
   - To search: <tool_call>["query1", ...]</tool_call>
     (System returns top-5 snippets per query)
   - To answer: <answer>["A", "B"]</answer>

Both <tool_call> and <answer> content must be
a JSON list of strings.

Question: {Question here}
```

Figure 5: The standard system prompt template.

B Datasets

To rigorously evaluate the diverse exploration and complex reasoning capabilities of the SPADER framework, we select four widely recognized benchmarks. Each dataset presents unique challenges in terms of logical constraints, long-tail entity retrieval, and multi-hop dependency:

- **QAMPARI** (Amouyal et al., 2023): An open-domain question answering (ODQA) benchmark specifically designed for questions that have many answers spread across multiple paragraphs. Unlike standard datasets that focus on a single answer from a single paragraph, every question in QAMPARI has at least 5 answers, averaging 13 answers per question. It tests a system's ability to retrieve and read a large number of passages from a corpus to generate exhaustive lists of entities.
- **Mintaka** (Sen et al., 2022): A large-scale, multilingual question answering dataset containing 20,000 question-answer pairs elicited in English and translated into eight other languages. A key differentiator is its focus on naturalness; rather than auto-generating queries, it features complex questions naturally elicited from crowd workers. It spans 8 distinct complexity types: count, comparative, superlative, ordinal, multi-hop, intersection, difference, and yes/no questions.
- **WebQSP** (Yih et al., 2016): A Knowledge Base Question Answering (KBQA) dataset that provides full SPARQL semantic parses for 4,737 questions from the original WebQuestions dataset. Rather than being designed for open-book text retrieval, it is explicitly grounded in Freebase to evaluate the value of training models on intermediate logical forms (semantic parses) rather than just final question-answer pairs.
- **QUEST** (Malaviya et al., 2023): A retrieval benchmark comprising 3,357 natural language queries that test a model's ability to handle implicit set operations, such as intersection, union, and difference. Each query maps to an exhaustive answer set of 2 to 20 Wikipedia entities. It challenges retrieval systems to match multiple constraints mentioned in a query with corresponding textual evidence in documents, without relying on structured knowledge bases.

C Implementation Details

Framework and Infrastructure. Our training pipeline is built upon the Verl framework (Sheng et al., 2025) for Reinforcement Learning optimization, utilizing sglang as the high-throughput inference engine for both training-time rollout sampling and evaluation. All experiments are conducted on a single node equipped with 8 NVIDIA A100 GPUs.

Training Configurations and Hyperparameters. The unified training set consists of 12,000 instances, including 6,000 from QAMPARI and 2,000 each from Mintaka, WebQSP, and QUEST. We train the models for 1 epoch, resulting in 375 optimization steps with a global batch size of 32. For gradient stability, we set the mini-batch size to 32 and micro-batch size to 4. We employ the AdamW optimizer with a learning rate of 1e-5 and a linear warmup ratio of 0.05. To facilitate step-wise policy optimization, our SPADER framework samples a group of $G = 16$ parallel trajectories for each prompt. To maintain policy stability, the KL-divergence penalty coefficient β_{KL} is fixed at 0.001. For advantage estimation, We set the discount factor γ to 0.9 to balance immediate rewards

Methods	QAMPARI			Mintaka			WebQSP			QUEST		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
SPADER	**0.474**	**0.312**	**0.343**	**0.625**	**0.626**	**0.611**	**0.423**	**0.409**	**0.388**	**0.233**	0.109	**0.127**
w/o SPA	0.404	0.268	0.285	0.591	0.566	0.569	0.380	0.343	0.336	0.184	**0.110**	0.117
w/o Novelty Premium	0.309	0.259	0.239	0.344	0.505	0.373	0.250	0.341	0.256	0.092	0.095	0.083
w/o Info Gain	0.399	0.224	0.266	0.340	0.425	0.364	0.345	0.414	0.359	0.164	0.074	0.094

Table 4: Full ablation results of SPADER on Qwen3-8B. "w/o" denotes removing one component from SPADER.

and long-term credit assignment.

Reward and Generation Settings. During the reinforcement learning phase, we use a temperature of 1.0 and *top_p* of 1.0 for diverse trajectory generation. The Diversity-Aware Exploration Reward is configured with a horizontal information gain weight $\alpha = 1.0$ and a vertical novelty premium weight $\beta = 4.0$. To prevent redundant tool calls and encourage efficiency, a tool-use cost of 0.01 is applied per search call.

Entity Normalization and Step-level Matching. For step-level reward settlement, we process the complete trajectory after each rollout. At decision step t, we take the returned tool-response snippets and determine step-level hits by matching the ground-truth entity set against these snippets after deterministic normalization, including lowercasing, Unicode normalization, punctuation and article removal, and whitespace collapse. Newly discovered entities at step t are the step-level hits that are not in the cumulative matched set from earlier steps.

We acknowledge that this string-based protocol may miss some lexical variants. However, this mainly introduces false negatives, so the induced bias primarily lowers absolute step-level rewards rather than inflating them. At the same time, because the same matching procedure is applied to all trajectories within a group, these misses are often shared across peers and therefore have limited effect on step-wise relative comparison under the group baseline. More robust matching remains an important direction for future investigation.

D Full Ablation Results

For completeness, we report the full ablation results with Precision (P), Recall (R), and F1 across all four datasets.

E Case Study

This section analyzes QAMPARI trajectories sampled from the Qwen3-8B backbone. Cases 1–3 use the same prompt to compare ReAct, SPADER, and the SPADER ablation without the information gain. Case 4 uses a different prompt to illustrate an ambiguity-driven mismatch between natural-language intent and evaluation target.

Case 1 and Case 2: Early Stop vs. Iterative Expansion. Case 1 and Case 2 share the same 75-answer prompt on Luzerne County municipalities, as shown in Figure 6 and Figure 7. In Case 1, ReAct exposes a useful decomposition signal in the first search action, including city, borough, and township cues, but terminates immediately after three calls and returns only 7 entities. Precision is not extremely poor, yet recall drops to 0.03, indicating a typical under-exploration pattern where coarse evidence is not converted into broader coverage.

In Case 2, SPADER does not only use more search actions, but also organizes them with a clearer progression. Step 1 builds a seed set, Step 2 branches by municipality type, and Step 3 shifts to a completeness-oriented sweep. Although later snippets become noisier, the accumulated entity set still expands from 7 to 42 predicted items, and F1 rises from 0.05 to 0.24.

Case 3: Ablation Dynamics Without Information Gain. Case 3 is shown in Figure 8. After removing the information gain component from the Diversity-Aware Exploration Reward, the early phase still discovers valid municipality names. However, after this short productive window, the trajectory enters a long low-yield regime in which many additional search actions produce almost no new entities.

The trace suggests a specific failure pattern: query templates keep changing lexically, while re-

turned evidence remains semantically repetitive, dominated by county-level background text and previously seen names. Without a strong incentive for incremental useful gain, the policy becomes less sensitive to diminishing returns and delays termination. The final outcome, 34 calls with 5 predictions and F1 of 0.05, is more consistent with search-action drift plus poor stopping than with a single catastrophic tool error.

Case 4: Ambiguity Under Broad Intent. Case 4 is shown in Figure 9. The prompt "Who served the country of Australia?" is semantically broad and can naturally be interpreted as political leaders or senior public officials, while the benchmark target set follows a different entity scope. Under this ambiguity, the first search action anchors the trajectory to a leadership interpretation that is semantically plausible but label-misaligned.

The resulting zero-overlap outcome should be interpreted cautiously. It reflects both policy behavior and a task-specification mismatch between natural-language intent and evaluation target. In this example, additional search depth, 9 calls and 37 predicted entities, does not resolve the mismatch once the initial interpretation diverges from the annotation scope.

Overall Takeaway. Across these examples, we observe two practical patterns. When intermediate evidence can be transformed into targeted follow-up search actions, SPADER tends to improve coverage over shallow one-step behavior. At the same time, some low-score cases also show a clear boundary, when prompt intent is broad or ambiguous, early intent misalignment can dominate the final error.

ReAct: Early Stop

Prompt: *What municipalities are located in Luzerne County, Pennsylvania?*
Ground Truth: total=75. Sample: ["Wilkes-Barre city", "Hazleton city", "Kingston borough", "Hanover Township", "Nanticoke city", "Plains Township", "Hazle Township", "Butler Township", ...].

Step 1
search: ["municipalities in Luzerne County Pennsylvania", "official list of municipalities Luzerne County PA", "Luzerne County PA cities towns boroughs"]
tool response: ["Luzerne County ... contains one of the highest numbers of independently governed municipalities in Pennsylvania ..."; "Bear Creek Township is a township in Luzerne County, Pennsylvania ..."; "Plains Township is a township in Luzerne County, Pennsylvania ..."; etc.]
new entities: ["Bear Creek Township", "Plains Township", "Wilkes-Barre", "Pittston", ...]
next-step rationale: county-level list should be expanded by municipality type, but the trajectory ends early.

Step 2
answer: **[Final Answer]** ["Wilkes-Barre", "Pittston", "Harveys Lake", "Jeddo", "Bear Creek", "Wilkes-Barre Township", "Plains Township"]

Metadata: predicted_count=7, precision=0.29, recall=0.03, f1=0.05, call_count=3.

Figure 6: ReAct: early stop after one search round leads to severe under-coverage.

SPADER: Iterative Expansion with Information Gain

Prompt: *What municipalities are located in Luzerne County, Pennsylvania?*
Ground Truth: total=75. Sample: ["Wilkes-Barre city", "Hazleton city", "Kingston borough", "Hanover Township", "Nanticoke city", "Plains Township", "Hazle Township", "Butler Township", ...].

Step 1
search: ["list of towns in Luzerne County PA", "municipalities Luzerne County Pennsylvania", "cities boroughs Luzerne PA"]
tool response: ["Plains Township is a township in Luzerne County, Pennsylvania ..."; "Hazle Township is a township in Luzerne County, Pennsylvania ..."; "Luzerne County ... has a very large number of municipalities ..."; etc.]
new entities: ["Hazle Township", "Plains Township", "Bear Creek Township", "Wilkes-Barre", "Pittston", "Hazleton", ...]
next-step rationale: first-hop evidence is township-heavy; next search should explicitly decompose by *cities / boroughs / townships*.

Step 2
search: ["list of cities in Luzerne County PA", "boroughs in Luzerne County Pennsylvania 2023", "townships Luzerne County Pennsylvania names"]
tool response: ["Plains Township ... township on the outskirts of Wilkes-Barre ..."; "Hazle Township ... township in Luzerne County ..."; "Luzerne County ... municipalities by local-government type ..."; etc.]
new entities: ["Foster Township", "Butler Township", "Conyngham", "Sugarloaf Township", "Black Creek Township", "Harveys Lake", "Jeddo", ...]
next-step rationale: type-specific queries recover many new names; now switch to completeness-oriented township inventory and by-type consolidation.

Step 3
search: ["list of townships in Luzerne County PA 2023", "municipalities in Luzerne County PA by type", "townships in Luzerne County Pennsylvania with names"]
tool response: ["Luzerne County ... administrative history and county-level references ..."; "state of Pennsylvania and the county seat of Luzerne County ..."; "Luzerne County ... large municipal count indicates remaining uncovered entities ..."; etc.]
new entities: ["Dallas Township", "Huntington Township", "Franklin Township", "Jackson Township", "Lake Township", "Lehman Township", ...]
next-step rationale: despite noisier snippets, prior accumulated entity set keeps growing and supports a broader final consolidation.

Step 4
answer: **[Final Answer]** ["Hazle Township", "Plains Township", "Bear Creek Township", "Wilkes-Barre", "Pittston", "Hazleton", "Luzerne", "Kingston", "Nanticoke", "West Hazleton", "Foster Township", "Butler Township", "Conyngham", "Sugarloaf Township", "Black Creek Township", "Harveys Lake", "Jeddo", "Wilkes-Barre Township", "Dallas Township", "Huntington Township", "Franklin Township", "Jackson Township", "Lake Township", "Lehman Township", ...]

Metadata: predicted_count=42, precision=0.33, recall=0.19, f1=0.24, call_count=9.

Figure 7: SPADER: incremental exploration expands coverage step by step.

SPADER without Information Gain: From Useful Start to Search-Action Drift

Prompt: *What municipalities are located in Luzerne County, Pennsylvania?*
Ground Truth: total=75. Sample: ["Wilkes-Barre city", "Hazleton city", "Kingston borough", "Hanover Township", "Nanticoke city", "Plains Township", "Hazle Township", "Butler Township", ...].

Step 1
search: ["What municipalities are in Luzerne County PA", "List of towns in Luzerne County Pennsylvania"]
tool response: ["Luzerne County ... has an exceptionally high number of municipalities ..."; "Bear Creek Township is a township in Luzerne County ..."; "Plains Township is a township in Luzerne County ..."; etc.]
new entities: ["Bear Creek Township", "Plains Township", "Dallas", ...]
next-step rationale: early responses contain valid municipality clues.

Step 2–4 (transition)
search: ["List of cities and boroughs in Luzerne County PA", "Municipalities in Luzerne County Pennsylvania 2023"]; ["Luzerne County cities list PA", "Luzerne County boroughs PA"]; ["Luzerne County townships PA", "Luzerne County towns PA"]
tool response: ["Dallas is a borough in Luzerne County, Pennsylvania ..."; "Bear Creek Township is a township in Luzerne County ..."; "County-level background lines increasingly repeat with limited new entities ..."; etc.]
new entities: ["Dallas", "Newport Township", "Huntington Township", ...]
next-step rationale: signal begins to dilute; many search calls return county facts rather than new municipality names.

Step 5–16 (drift / noisy exploration phase)
search: ["Luzerne County municipalities PA 2023", "List of all towns in Luzerne County PA"]; ["Luzerne County municipal list 2023", "PA Luzerne County incorporated areas"]; ["Luzerne County city and borough list", "List of all Luzerne County townships"]; ["Luzerne County municipal list final 2023", "PA Luzerne County all incorporated areas complete"]; ...
tool response: ["Huntington Township is a township in Luzerne County ..."; "Dallas Township is a township in Luzerne County ..."; "Repeated county-level boilerplate dominates (e.g., topography/history) with weak incremental gain ..."; etc.]
new entities: [none]
next-step rationale: query templates keep mutating but mostly recycle existing evidence and fail to converge.

Step 17
answer: **[Final Answer]** ["Wilkes-Barre", "Hazleton", "Plains Township", "Bear Creek Township", "Dallas"]

Metadata: predicted_count=5, precision=0.40, recall=0.03, f1=0.05, call_count=34.

Figure 8: SPADER without information gain: search actions drift into repetitive, low-gain calls with degraded stopping behavior.

SPADER Failure Case: Broad Intent Causes Entity-Type Drift

Prompt: *Who served the country of Australia?*

Ground Truth: total=63. Sample: ["John Murray", "Cecil Foott", "Dominic McCarthy", "Edmund Lind", "Jack Studdert", "Ringer Edwards", "Maurice Fergusson", "David Johnston", ...].

Step 1
search: ["list of Australian Prime Ministers", "Governors-General of Australia list", "people who served as leaders in Australia government"]
tool response: ["List of prime ministers of Australia ..."; "Historical rankings of prime ministers of Australia ..."; "Records of prime ministers of Australia ..."; etc.]
new entities: ["Edmund Barton", "Andrew Fisher", "Joseph Cook", "John Curtin", ...]
next-step rationale: after a leader-heavy first pass, the model broadens query scope to additional public-service roles.

Step 2
search: ["list of Australian state premiers", "Australian Chief Justices list", "notable Australian public servants in government"]
tool response: ["List of West Australian Football League premiers ..."; "West Australian State Premiership ..."; "Premier of Tasmania ..."; etc.]
new entities: [none]
next-step rationale: retrieved content drifts away from target entity type, but the model continues broad government-style queries.

Step 3
search: ["list of Australian Cabinet Secretaries", "notable Australian public officials in government history", "senior public servants who served in Australian federal government"]
tool response: ["List of West Australian Football League premiers ..."; "West Australian State Premiership ..."; "Premier of Tasmania ..."; etc.]
new entities: [none]
next-step rationale: repeated off-target evidence is accumulated, and the model consolidates a leader-centric final list.

Step 4
answer: **[Final Answer]** ["Edmund Barton", "Andrew Fisher", "Joseph Cook", "John Curtin", "Ben Chifley", "Gough Whitlam", "Bob Hawke", "Paul Keating", "John Howard", "Kevin Rudd", ...]

Metadata: predicted_count=37, precision=0.00, recall=0.00, f1=0.00, call_count=9.

Figure 9: broad-intent ambiguity: a plausible but misaligned interpretation leads to low overlap with the target set.