

Action-Level Credit Assignment for Retrieval-Augmented Reasoning

Laura Perez-Beltrachini and Mirella Lapata

Institute for Language, Cognition and Computation

School of Informatics, University of Edinburgh

10 Crichton Street, Edinburgh EH8 9AB

{lperez,mlap}@inf.ed.ac.uk

Abstract

Complex question answering over large document collections requires models to identify missing information, issue sub-queries, and iteratively gather evidence before producing an answer. Reinforcement learning offers a natural framework for training such behaviour, but standard GRPO-style training provides only a sparse final-answer reward and assigns the same advantage to every token in a rollout. We introduce **PACE-GRPO** (Potential-based Action Credit Estimation GRPO), a variant of GRPO that combines potential-based reward shaping with action-level credit assignment. PACE-GRPO computes a state potential from inexpensive retrieval-derived signals, including retrieval utility, coverage, and passage diversity, and uses potential differences to provide step-level feedback. Unlike prior intermediate-reward methods, it requires no gold evidence annotations, teacher traces, reward model, or additional forward passes. Across six multi-hop QA benchmarks, PACE-GRPO learns a tool-use policy that selectively invokes the more expensive retrieval operation only when it is beneficial. This adaptive tool use yields consistent improvements over standard GRPO and strong search-augmented baselines.¹

1 Introduction

Complex question answering (QA) over large document collections requires more than a single retrieval step; a model must reason about what it knows, identify gaps, formulate targeted sub-queries, and iteratively gather evidence before synthesising a final answer. Current moderate-size open-weight LLMs have made significant strides in reasoning and tool use, yet they consistently fail to execute the sustained sequence of retrieval and reasoning steps required to solve such tasks reliably (Trivedi et al., 2023; Song et al., 2025). Fine-tuning

via reinforcement learning (RL) can help models grasp the structure of multi-step retrieval reasoning, but performance plateaus well below what oracle retrieval conditions suggest is achievable.

Several interrelated limitations explain this gap. First, RL training for QA is typically governed by a sparse outcome reward — a single correctness signal at the end of a potentially long reasoning chain — which provides little guidance for improving intermediate retrieval and reasoning decisions (Wang et al., 2024; Wei et al., 2025b; Wen et al., 2024). Process reward models have been proposed to address this (He et al., 2025; Goldie et al., 2025), but they typically assign a uniform score across all intermediate steps and require large, expensive reward models. Second, LLM-based RL agents are known to suffer from limited exploration (Amani et al., 2026), converging quickly to repetitive retrieval strategies that fail to discover more informative evidence paths. Finally, and perhaps most specific to retrieval-augmented settings, there is a persistent generator-retrieval mismatch: sub-queries generated by the LLM may be poorly aligned with the retriever’s query distribution (Meng et al., 2026), yielding low-quality passages that undermine downstream reasoning regardless of the quality of the generator itself.

Recent work has begun tackling some of these issues in isolation. Qian and Liu (2025) reward intermediate passage coverage but require gold-annotated training data. Xie et al. (2026) supervise information gain at each retrieval step by comparing answer probabilities before and after retrieval, but this doubles the number of forward passes during training. Both approaches also leave the retriever fixed and do not adapt the retrieval strategy to the difficulty or nature of each sub-question.

In this work, we take a different approach by exploiting cheap, direct signals from a *rich retrieval environment* — signals that require no additional model forward passes, no gold passage annotations,

¹Our code and data are in github.com/lauhaide/pace.

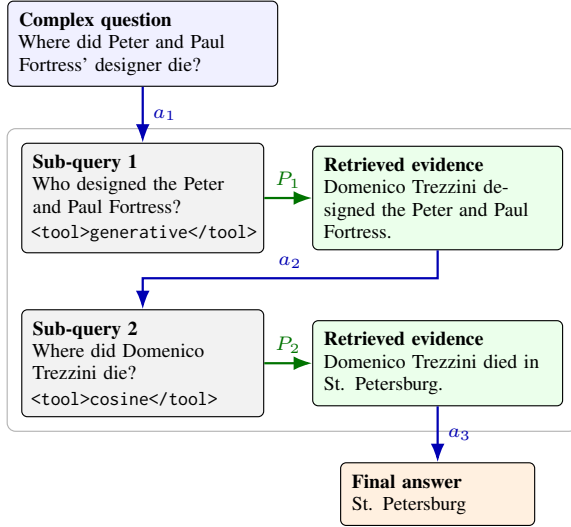


Figure 1: A question from MuSiQue (Trivedi et al., 2023) is answered through sub-queries and tool selection. Blue edges denote agent actions; green edges denote retrieved passage sets, appended to the agent context for the next action. The agent first uses a deep generative retriever to identify the relevant entity, then a cosine similarity based retriever for named-entity lookup.

and no strong teacher model. We frame retrieval as *tool selection*: at each step, the agent chooses not only *what* to search for but *which retrieval tool* to invoke from a set $\mathcal{T}$ of available tools with different cost and capability profiles (Schick et al., 2023; Yao et al., 2023; Chan et al., 2024). In our setting, these are a fast cosine similarity retriever suited to factoid lookups and a generative retriever suited to more complex information needs. By interacting with the environment, the agent learns to match the retrieval tool to the sub-question at hand, supervised by the observed utility gain across tools. Figure 1 illustrates this for a multi-hop question.

We propose **PACE-GRPO** (Potential-based Action Credit Estimation GRPO), a novel variant of GRPO (Shao et al., 2024; DeepSeek-AI, 2025) that combines potential-based reward shaping (Ng et al., 1999) with action-level credit assignment. Rather than assigning a single outcome reward uniformly across all tokens in a rollout, PACE-GRPO computes a *state potential* at each step from three inexpensive retrieval-based measures: retrieval utility, passage coverage of a silver reference set built offline, and within-rollout passage diversity. The difference in consecutive state potentials forms a shaping term that is used to modulate the GRPO advantage at the level of individual actions (Wei et al., 2025b; Wen et al., 2024), providing an informative potential-based step-level credit signal (Ng

et al., 1999) without requiring an external reward model or gold intermediate supervision.

We evaluate on six challenging multi-hop QA benchmarks requiring sustained multi-step reasoning. Our results show that PACE-GRPO outperforms both outcome-only GRPO and competitive baselines with intermediate supervision in answer quality, while learning a balanced tool-use policy that avoids over-reliance on the expensive generative retriever. These results demonstrate that inexpensive signals from a rich retrieval environment can substitute for costly teacher supervision, leading to both more accurate and more efficient retrieval-augmented reasoning.

2 Related Work

Retrieval-augmented Reasoning Complex QA over large corpora has motivated methods that interleave reasoning with retrieval, rather than relying on a single retrieve-then-read step. IRCoT (Trivedi et al., 2023) showed that chain-of-thought reasoning can guide multi-step retrieval, and subsequent work has explored query refinement, chain-of-retrieval, collaborative RAG, and search-during-reasoning frameworks (Chan et al., 2024; Xu et al., 2025; Wang et al., 2025; Shi et al., 2025). These methods demonstrate the importance of iterative evidence acquisition, but typically treat retrieval as a fixed operation conditioned on generated queries.

RL for Search-augmented LLMs Recent work has used reinforcement learning to improve the search behaviour of LLM agents. R1-Searcher (Song et al., 2025) incentivises search use through outcome-based rewards, while Scent (Qian and Liu, 2025) and TIPS (Xie et al., 2026) introduce intermediate supervision for search-augmented reasoning. Scent requires gold passage sets, while TIPS requires additional forward passes per retrieval step. Goldie et al. (2025) use synthetic data generation combined with multi-step RL to improve reasoning and tool use, but rely on a strong teacher model to generate training trajectories. In contrast, our method derives step-level feedback directly from inexpensive retrieval-environment signals, without gold intermediate annotations, teacher traces, or extra answer-probability computations. Concurrent work also targets retrieval efficiency alongside accuracy: Ji et al. (2025) use a multi-factor curriculum reward that penalises redundancy, while Java et al. (2026) apply RL to prune search depth adaptively.

Reward Shaping and Credit Assignment

Outcome-only RL provides a sparse correctness signal for long reasoning trajectories, making it difficult to assign credit to intermediate search and reasoning decisions. Process rewards and turn-level credit assignment have been proposed to address this problem (Wang et al., 2024; Goldie et al., 2025; Wei et al., 2025b; Wen et al., 2024), and GRPO-style objectives have become a standard choice for efficient LLM reinforcement learning (Shao et al., 2024; DeepSeek-AI, 2025). Our approach is closest in spirit to turn-level reward design, but differs in grounding the intermediate signal in potential-based reward shaping (Ng et al., 1999). More recently, Tan et al. (2025) propose entropy-weighted token-level credit assignment, but use policy entropy as a weighting signal rather than grounding the credit signal in retrieval environment feedback.

Adaptive Retrieval and Tool Selection LLMs can be trained or prompted to use external tools (Schick et al., 2023; Yao et al., 2023), and retrieval-augmented systems increasingly expose models to richer search operations. Work on query refinement and reasoning-aware retrieval improves the quality of generated queries or retriever representations (Chan et al., 2024; Das et al., 2025; Shao et al., 2025). Our focus is complementary: we do not assume a single fixed retriever, but treat retrieval as a tool-selection problem. Concurrent work also trains LLMs to select among multiple retrieval tools through RL (Luo et al., 2025), but relies on trajectory-level outcome rewards (including number of called tools) rather than action-level credit assignment; tool-use knowledge is mostly acquired in an initial standard fine-tuning stage with rollouts from an expert.

3 The PACE-GRPO Framework

We formulate complex question answering as a reinforcement learning task in which an LLM-based agent learns to answer questions by interacting with a retrieval environment. We first describe the environment — its state space, action space, and feedback mechanisms — and then present the learning algorithm that trains the agent within it. Throughout this section, we use k for environment/action steps, i for token positions within an action, and j for rollout indices within a GRPO group. Thus s_k denotes the state after action a_k , whereas $a_{k,i}$ denotes the i -th token of action a_k . Figure 2 gives

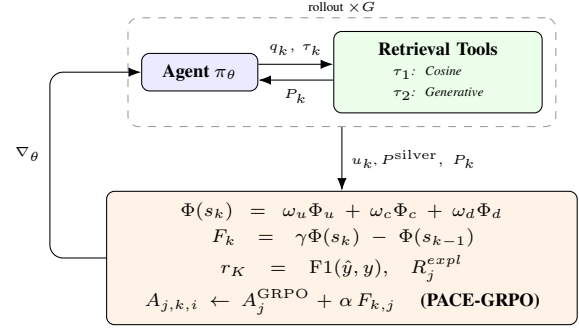


Figure 2: PACE-GRPO training loop. The agent π_θ interacts with the retrieval environment over up to K steps, generating G rollouts per question. Utility scores u_k returned by the retrieval tools, passages P_k , and offline silver passages P^{silver} feed into the state potential $\Phi(s_k)$; discounted potential differences give action-level signals that modulate GRPO advantages. Passages $\Phi(s_k)$ are used to compute an exploration reward R_j^{expl} .

an overview of the full training loop.

3.1 The Retrieval Environment

Let π_θ be an LLM-based QA agent whose goal is to generate answer $\hat{y}$ to user question x by interleaving reasoning with information retrieval from an external document database $\mathcal{C}$. The agent interacts with the environment over action steps $k \in \{1, \dots, K\}$. It starts in an initial state s_0 containing the task instruction and question x . At each step k , the agent receives the current state s_{k-1} , takes action a_k , and transitions to a new state s_k . Figure 1 illustrates this process; we define the action space and transition dynamics formally below.

The agent can take two types of **actions**: SEARCH or ANSWER. Each action consists of a brief reasoning span followed by a structured tag:

SEARCH `<search> $q_k$ </search>` where q_k is a sub-question targeting a specific piece of missing information, together with a tool selection tag `<tool> $\tau_k$ </tool>`, where $\tau_k \in \mathcal{T}$ specifies which retrieval tool to invoke.

ANSWER `<answer> $\hat{y}$ </answer>`, where $\hat{y}$ is the final answer to question x .

When the action is ANSWER, the resulting state is the terminal state s_K from which $\hat{y}$ is extracted.

Information needs vary in complexity: some sub-questions target known named entities or facts that can be resolved by a cheap similarity function (e.g., *Schwarzenegger's date of birth*), while others require deeper semantic analysis (e.g., *Law*

signed by Schwarzenegger as Governor of California). Rather than fixing a single retrieval strategy, we equip the environment with two retrieval tools, each with a different cost and capability profile, and train the agent to select the appropriate tool for each sub-question. In our experiments, we define the set $\mathcal{T}$ of tools as:

Cosine retriever (τ_1) performs a cosine similarity search and returns the top- N passages P_k from $\mathcal{C}$. It is fast and well suited to sub-questions involving specific named entities or exact phrases.

Generative retriever (τ_2) is a dense retriever that first retrieves a larger candidate set of $M \gg N$ passages, then re-ranks them using a more expressive re-ranker $\mathcal{R}''$, returning the top- N passages P_k'' . It is more expensive but better suited to sub-questions requiring deeper semantic matching.

The agent learns when τ_2 is worth its additional cost through interaction with the environment. If the agent selects τ_2 but the re-ranked passages do not improve on those returned by τ_1 , i.e., $\text{mean}(u_k'') \leq \text{mean}(u_k) + m$, where u_k and u_k'' are the passage utility scores for P_k and P_k'' and m is a margin, the environment penalises the agent by returning a fraction of the retriever score. If the agent generates an invalid tool label $\tau_k \notin \mathcal{T}$, the environment generates a negligible retrieval score with the message: The tool selection is incorrect; please select a valid retrieval tool, either τ_1 or τ_2 . This setup can be easily augmented with additional tools; for example, a structured knowledge base query or a web search engine can be added to $\mathcal{T}$ without modifying the learning framework.

If the search action is successful, the retrieved passages P_k are appended within `<information>` P_k `</information>` tags to form the environment observation O_k . If retrieval fails, the environment returns `<information>`No results`</information>`. And the new state is $s_k = (s_{k-1}, a_k, O_k)$. After at most K actions, the agent reaches a terminal state $s_K = (O_0, a_1, O_1, \dots, a_K, O_K)$, where O_0 encodes the task instruction and question x . This trajectory constitutes a *rollout*.

3.2 Defining the State Potential

Following standard practice (Song et al., 2025; Qian and Liu, 2025; Xie et al., 2026), the environment provides a non-zero terminal reward r_K

at the final state s_K , computed as the F1-score of $\hat{y}$ against the gold answer y . In addition, we define a *state potential* $\Phi(s_k)$ that evaluates the quality of each intermediate state using only cheap, information-theoretic signals from the retrieval environment. Intuitively, a state is good if it incorporates relevant new information that closes the gap towards the goal answer. We define Φ as a weighted combination of four measures.

Retrieval score Φ_u . Given passages P_k retrieved for sub-question q_k with utility scores $u_k = \{u_1, \dots, u_{|P_k|}\}$:

$$\Phi_u(s_k) = \max(u_k). \quad (1)$$

This captures how well the selected tool and generated sub-query align with both the retriever’s query distribution and the content of $\mathcal{C}$, providing implicit feedback on the quality of tool selection.

Coverage score Φ_c . As the agent accumulates passages across steps it should move closer to the information needed to answer correctly (Qian and Liu, 2025). Since we do not assume training data with gold intermediate passages, we approximate this by constructing *silver passages* P^{silver} offline for each training instance, by retrieving and re-ranking passages using the concatenated question and gold answer (x, y) as a query. The coverage score after action k is:

$$\Phi_c(s_k) = \log(|P_{1\dots k} \cap P^{\text{silver}}| + 1). \quad (2)$$

Details on silver passage quality are provided in Appendix B.

State diversity Φ_d . RL agents for QA tend toward *linear exploration*: when a sub-question q_k fails, at step $k + 1$ the agent often generates a reformulation q_{k+1} , retrieving similar or identical passages regardless of which tool is selected. To discourage such inefficient policies, we measure the diversity of the passages accumulated up to step k as the proportion of unique passages among all retrieved passages:

$$\Phi_d(s_k) = \frac{|\{p \mid p \in P_{1\dots k}\}|}{|P_{1\dots k}|}. \quad (3)$$

A low value indicates that the agent is repeatedly retrieving the same passages — a sign of redundant tool calls or repetitive query reformulation — and is penalised accordingly, incentivising the agent to

explore new retrieval paths rather than converging on the same evidence.

Our combined state potential score is computed as the weighted sum of these state scores:

$$\Phi(s_k) = \omega_u \Phi_u(s_k) + \omega_c \Phi_c(s_k) + \omega_d \Phi_d(s_k). \quad (4)$$

3.3 Sample Diversity

LLM-based RL agents are known for limited exploration across rollouts (Amani et al., 2026; Zhang et al., 2026), and measuring semantic diversity across G rollouts is typically expensive (Li et al., 2026). In our retrieval environment, however, we can assess whether two rollouts explore different search paths cheaply by comparing their retrieved passage sets: if the sets differ, the agent invoked different tools or generated different sub-questions, indicating genuine exploration. We formalise this as the average state diversity of rollouts g^j in a group G :

$$R^{expl} = -\frac{1}{K-1} \sum_{k=1}^{K-1} \log \frac{1 + \sum_{\ell \neq j} \mathbb{1}[P_k^j \subseteq P_{1\dots k}^\ell]}{G}. \quad (5)$$

The add-one term is equivalent to counting rollout j itself as a match, ensuring that the logarithm is always well-defined. Thus, R^{expl} is largest when the passages retrieved by rollout j are not contained in any other rollout’s accumulated evidence, and smallest when the same passage set is replicated across all rollouts. A high value indicates that the passage set retrieved in rollout j at step k is rarely replicated in other rollouts, rewarding exploration across the group.

3.4 Learning from the Retrieval Environment

Training the agent requires optimising a policy over a token-level Markov Decision Process (MDP), where actions are sequences of tokens and rewards are provided by the retrieval environment.

The agent generates each action token by token. If $a_k = (a_{k,1}, \dots, a_{k,|a_k|})$, then

$$\pi_\theta(a_k | s_{k-1}) = \prod_{i=1}^{|a_k|} \pi_\theta(a_{k,i} | a_{k,<i}, s_{k-1}). \quad (6)$$

Standard PPO (Schulman et al., 2017) is usually defined over the flattened token trajectory. Let n index generated tokens in the full rollout, and let $s(n)$ denote the state/history available before token n is generated. PPO uses an advantage estimate

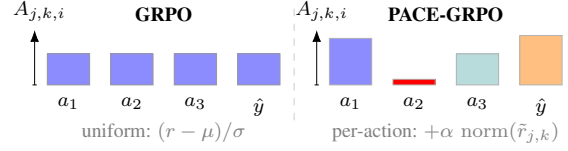


Figure 3: Standard GRPO assigns a uniform advantage to every token in a rollout (left); PACE-GRPO (right) modulates it per action via the signal $F_{j,k}$: a productive search a_1 earns a high advantage, an incorrect tool call a_2 is penalised (red bar), and the correct answer $\hat{y}$ is rewarded.

$A_n = G_n - V_\phi(s(n))$, where

$$G_n = \sum_{m=n}^N \gamma^{m-n} r_m. \quad (7)$$

Here N is the total number of generated tokens in the rollout. GRPO (Shao et al., 2024) removes the learned value baseline by sampling a group of G rollouts and normalising their terminal rewards within the group, i.e., $A_n^{GRPO} = (r_j - \mu)/\sigma$ where r_j is the outcome reward for rollout j and μ and σ are the mean and standard deviation of outcome rewards within its GRPO group. Thus, in standard GRPO, every generated token in rollout j receives the same group-relative advantage. This makes GRPO considerably more efficient than PPO and has become the standard choice for LLM reinforcement learning (DeepSeek-AI, 2025).

Despite its efficiency, standard GRPO has two limitations that are problematic in our setting. First, it relies solely on a *sparse terminal reward*: the agent receives a single correctness signal only after producing its final answer, with no feedback on the quality of intermediate retrieval steps or tool selections. In long multi-hop reasoning chains this makes it very difficult to identify which search actions were responsible for success or failure. Second, it assigns the *same advantage to every token in a rollout*, conflating the contribution of good and bad actions within the same trajectory (Wang et al., 2024; Wei et al., 2025b; Wen et al., 2024; Xie et al., 2026). A rollout that retrieves highly relevant passages early but then degenerates into repetitive queries is treated identically to one that makes steady progress throughout.

We address both limitations with **PACE-GRPO** (Potential-based Action Credit Estimation GRPO), which augments GRPO with two ideas: a potential-based progress signal that provides dense intermediate feedback, and action-level credit assignment

that differentiates the contribution of individual steps within a rollout (see Figure 2).

Following potential-based reward shaping (Ng et al., 1999), we define an action-level progress signal from differences in states’ potential. For a transition from s_{k-1} to s_k , the discounted potential-based shaping term is

$$F_k = \gamma \Phi(s_k) - \Phi(s_{k-1}). \quad (8)$$

In the original potential-based reward shaping framework, adding F_k to the reward preserves the set of optimal policies when the standard conditions hold, including constant terminal potentials (e.g., $\Phi(s_K) = 0$ for all terminal states).

Our use of this signal differs from classical reward shaping: we use it to modulate the GRPO advantage at the *action level* rather than to define a new MDP reward alone, and our potentials are retrieval-derived heuristics. We therefore do not claim strict policy invariance; the terminal $r_K = \text{F1}(\hat{y}_j, y)$ reward remains the primary GRPO objective. Our state potential is not defined for the ANSWER action a_K and terminal state s_K , i.e., $\Phi(s_K) = 0$. Following Xie et al. (2026), we assume the final shaping term to be $F_K = \gamma r_K - \Phi(s_{K-1})$; while in their case it enforces policy invariance, our term F_K modulates the answer segment advantage encouraging answer transitions with the largest improvement in F1.

For each action transition $s_{k-1} \rightarrow s_k$, we compute the discounted potential-based shaping term F_k . For every token i belonging to action a_k , the token-level advantage is modulated as

$$A_{j,k,i} \leftarrow A_j^{\text{GRPO}} + \alpha F_{j,k}, \quad (9)$$

where A_j^{GRPO} is the standard group-relative advantage assigned to rollout j , and α controls the strength of the action-level credit signal. The term $F_{j,k}$ is bounded by values of Φ and will be positive or negative depending on whether the action a_k results in a transition to a better or worse state.

Note that while $F_{j,k}$ assigns credits to actions, in PACE-GRPO the advantage is highly governed by the rollouts’ final rewards and their group normalisation A_j^{GRPO} . In a group, rollouts may exist which (all) have the same outcome while they differ in quality. To differentiate rollouts in a group in terms of the desired question answering behaviour, we combine the F1 reward with the averaged rollout potentials $\Phi_{\text{avg}_j} = \frac{1}{K-1} \sum_{k=1}^{K-1} \Phi(s_{j,k})$. To further strengthen diversity across rollouts we also

add the exploration reward defined in Section 3.3. Thus, the final rollout reward we use is:

$$R_j = \text{F1}(\hat{y}_j, y) + \lambda_\Phi \Phi_{\text{avg}_j} + \lambda_{\text{expl}} R_j^{\text{expl}} \quad (10)$$

4 Experimental Setup

Datasets We train on a 9,000-question sample from **MuSiQue** (Trivedi et al., 2023), which provides compositionally diverse multi-hop questions with explicit reasoning chains. We evaluate in-domain on a held-out MuSiQue sample, and out-of-domain on five benchmarks selected to probe complementary aspects of multi-hop reasoning:

- **HotpotQA** (Yang et al., 2018) is a large-scale Wikipedia-grounded benchmark with both bridge and comparison questions, widely used for direct comparison with prior work.
- **2WikiMultiHopQA** (Ho et al., 2020) tests generalisation beyond MuSiQue’s compositional style, covering four distinct reasoning patterns derived from structured Wikidata triples.
- **Bamboogle** (Press et al., 2023) stress-tests compositional generalisation with manually constructed questions designed to defeat surface-level retrieval.
- To assess the method’s generality, we additionally evaluate on **CRAG** (Yang et al., 2024) and **BrowseCompPlus** (Chen et al., 2026), two multi-hop benchmarks grounded on Web documents. The first one consists of natural user-like questions and knowledge base questions including tail entities. The second one contains complex questions requiring a larger number of hops and the ability to formulate strategic question decompositions and persistent search for difficult to find information. More details on these benchmarks are provided in Appendix A.

Evaluation Metrics We evaluate system output using the standard token-based metrics of Exact Match (EM) and F1. Since such metrics can underestimate answer correctness when valid responses differ lexically from the reference (Perez-Beltrachini and Lapata, 2025), we additionally report LLM-judged accuracy (Jdg). Specifically, we use GEMMA3-27B-INSTRUCT (Gemma Team et al., 2025) with an evaluation prompt based on that of Sun et al. (2024) to obtain binary correctness judgments. Details of the LLM evaluator are in Appendix C.

Method	MuSiQue		HotPotQA		2Wiki		Bamboogle		CRAG		BCPlus		Average		
	F1	Jdg	F1	Jdg	F1	Jdg	F1	Jdg	F1	Jdg	F1	Jdg	F1	Jdg	Steps
Closed-Book	.08	.07	.24	.29	.29	.31	.13	.10	.24	.29	.04	.02	.17	.18	—
Vanilla RAG	.12	.13	.43	.52	.13	.39	.27	.34	.31	.43	.05	.05	.22	.31	—
IRCoT	.13	.23	.41	.60	.30	.48	.37	.49	.24	.41	.02	.04	.24	.37	—
Zero-Shot	.18	.18	.37	.42	.36	.38	.51	.52	.36	.45	.05	.05	.31	.33	1.03
GRPO-R1	.38	.41	.62	.73	.72	.79	.61	.62	.41	.55	.09	.07	.47	.53	4.50
TIRESRAG-R1	.32	.36	.62	.73	.67	.76	.61	.63	.42	.54	.07	.09	.45	.52	3.60
MT-GRPO	.36	.40	.61	.71	.72	.77	.63	.66	.41	.55	.11	.10	.47	.53	5.51
LeTS	.40	.44	.65	.75	.71	.79	.66	.69	.42	.56	.10	.11	.49	.56	4.31
PACE-GRPO	.40	.45	.64	.76	.71	.81	.66	.68	.41	.59	.22	.23	.51	.59	5.00

Table 1: Results on six multi-hop QA benchmarks (QWEN3-4B-INSTRUCT; all systems retrained under an identical setup). All datasets except MuSiQue are out-of-domain. PACE-GRPO outperforms all baselines on every dataset and metric, matching or exceeding more heavily supervised methods while relying only on inexpensive retrieval-environment signals. Best per column in **bold**. Jdg: LLM-judge accuracy; Steps: average retrieval actions per question; ‘—’: not applicable; **BCPlus**: BrowseCompPlus.

Baselines and Comparison Systems We compare *PACE-GRPO* against three groups of baselines. The first group covers **prompt-only methods** that require no fine-tuning. *Closed-book* prompts the backbone model directly with no retrieved context, establishing the floor for parametric knowledge alone. *Vanilla RAG* issues a single retrieval call with the original question and feeds the top- N passages to the model. *Zero-shot* search uses a standard search-augmented QA prompt (Appendix C) and allows the model to issue an arbitrary number of sequential searches, but without any training signal. *IRCoT* (Trivedi et al., 2023) interleaves chain-of-thought reasoning with cosine retrieval, using each reasoning step to formulate the next retrieval query.

The second group covers **outcome-only RL methods**. *GRPO-R1* is a representative of several independently proposed methods (Song et al., 2025; Jin et al., 2025a; Chen et al., 2025) that fine-tune an LLM with GRPO using only a terminal F1 reward and a single fixed retriever; we train our own instance with identical hyperparameters for a controlled comparison.

The third group covers **RL methods with intermediate supervision**. *TIRESRAG-R1* (He et al., 2025) augments the outcome reward with two process rewards computed by a strong teacher model: one assesses whether the full trajectory (reasoning steps and retrieved passages) is sufficient to reach a correct answer, and the other evaluates the internal consistency of the reasoning chain. Additional rule-based signals reward reflective behaviour and penalise low-confidence steps, making this the most

heavily supervised baseline.

MT-GRPO (Wei et al., 2025b) instead uses a lightweight intermediate rule-based reward that assigns positive points to a retrieval tool-call action if the gold answer is contained in the retrieved passage set. We re-implement this approach following (Xie et al., 2026), intermediate action-level rewards are normalised (mean and variance) across steps of rollouts in a GRPO group and then combined with the group normalised outcome reward.

LeTS (Zhang et al., 2025) uses two lightweight rule-based process rewards that measure action-level diversity in successful rollouts and action alignment of unsuccessful rollouts to successful ones within a GRPO group. The optimal alignment is computed using the Kuhn-Munkres algorithm. LeTS normalises action-level rewards within each rollout and then combines them with the normalised outcome one. We re-implement LeTS following the description in (Zhang et al., 2025), given that no code is provided.

All fine-tuned baselines use the same backbone (QWEN3-4B-INSTRUCT), training set, retrieval corpus, and hyperparameters as PACE-GRPO for fair comparison; implementation details are given in Appendix B.

5 Results

PACE-GRPO Improves QA Performance Table 1 reports results for all baselines and comparison systems alongside PACE-GRPO (Table 11 in the Appendix shows EM values). The Closed-book baseline establishes how poorly the backbone copes with complex multi-hop questions without

any retrieval support: despite being a recent, capable small-scale LLM, it scores on average 17 F1 points across all benchmarks, confirming that the required knowledge cannot be retrieved from parameters alone. Prompt-based methods with retrieved context offer some gains but still remain below 31 F1 points. Vanilla RAG, which issues a single retrieval call with the original question, gets the biggest improvement over Closed-book on HotpotQA, where many questions can be resolved from a small number of passages (Perez-Beltrachini and Lapata, 2025). Iterative prompt methods obtain modest improvements over Vanilla RAG, suggesting that iterative retrieval without fine-tuning is insufficient for sustained multi-hop reasoning. The Zero-shot search prompt is the strongest prompt-only method, slightly better than IRCOT, benefiting from some explicit question decomposition. However, its overall performance is still low with a 1.03 average number of retrieval steps reflecting the backbone’s limited ability to plan and execute multi-step retrieval without explicit training.

All fine-tuned models improve substantially over the prompt-only baselines, but the three that provide no differentiated credit across actions perform almost identically: GRPO-R1 (outcome reward only), TIRESRAG-R1 (trajectory-level process reward broadcast uniformly), and MT-GRPO (rule-based action reward) lie within .02 average F1 of one another. They also generalise poorly to the out-of-domain benchmarks, particularly Bamboogle and BrowseCompPlus; this is most likely because their sparse terminal rewards and single fixed retriever leave intermediate retrieval decisions largely unsupervised. LeTS, which transfers action-level supervision from successful to non-successful rollouts within a group, is the strongest baseline on average. Its largest gain over GRPO-R1 is on Bamboogle, while it performs on par on 2Wiki and CRAG. Interestingly, on BrowseCompPlus it barely improves over GRPO-R1. Although step-wise self-supervision mitigates reward sparsity, the type of question decompositions learnt by LeTS are better suited to factual, template-like queries which do not meet the more open-ended information needs of BrowseCompPlus.

PACE-GRPO achieves the best overall performance across the six benchmarks. Although trained only on MuSiQue, it learns to produce both simple factual decompositions and complex contextualised sub-questions, and to condition its choice of retrieval tool on the difficulty of each (Appendix E).

Method	Average			
	EM	F1	Jdg	Steps
Zero-shot	.26	.35	.43	1.75
Naive-GRPO	.39	.49	.56	2.49
GRPO $_{\Phi_{avg}}$	.42	.52	.60	3.67
Φ MT-GRPO	.39	.48	.55	3.41
Φ LeTS	.37	.48	.57	3.22
PACE-GRPO	.41	.51	.59	5.00

Table 2: Reward-signal ablation for tool-use learning (QWEN3-4B-INSTRUCT). Naive-GRPO: terminal F1 reward only; GRPO $_{\Phi_{avg}}$: adds rollout-averaged potential as scalar process reward; MT-GRPO: potentials normalised across rollout steps; PACE-GRPO: action-level credit from potential differences. Jdg: LLM-judge accuracy; Steps: mean retrieval actions per question.

This allows PACE-GRPO to be efficient with questions answerable by direct lookup *and* to generalise to complex search-intensive QA tasks such as BrowseCompPlus.²

The Steps column in Table 1 reveals that PACE-GRPO issues more retrieval actions on average (5.00) than GRPO-R1 (4.50) and LeTS (4.31), yet achieves consistently better results, indicating that the additional steps reflect more deliberate evidence gathering rather than redundant querying. PACE-GRPO exceeds TIRESRAG-R1 despite the latter benefiting from a larger Qwen3-8B teacher model during training, a meaningful asymmetry in supervision cost.

PACE-GRPO Learns a Balanced Retrieval Tool-Use Policy In Table 2 we ablate the contribution of action-level credit assignment to retrieval tool selection by comparing PACE-GRPO against five GRPO variants that differ only in how environment feedback is incorporated. *Naive-GRPO* trains with the terminal F1 reward alone, relying entirely on this sparse outcome signal to discover when the generative retriever is worth invoking. GRPO $_{\Phi_{avg}}$ collapses the per-step potential scores into a single scalar by averaging across all steps in a rollout and adds this as a rollout-level process reward before group normalisation (c.f., Section 3.4); unlike PACE-GRPO, it therefore provides no differentiated signal across actions within a rollout.

²PACE-GRPO exhibits the largest F1-Jdg gap on CRAG (.41 vs .59). Manual inspection indicates this is largely a surface-form effect: answers to numerical and date questions are frequently correct but worded differently from the reference (e.g., *six* for 6, *3 December 1995* for *03-12-1995*), which token-overlap F1 penalises. This also accounts for its nominally lower CRAG F1 relative to LeTS despite the highest judge accuracy on that benchmark.

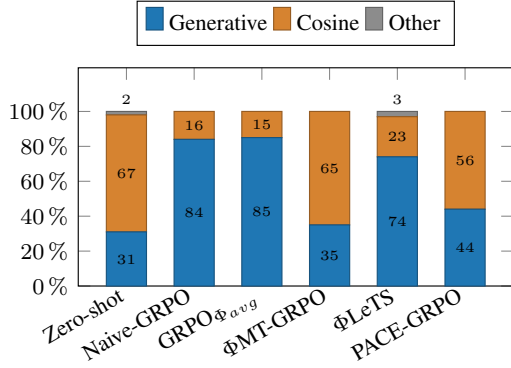


Figure 4: Retrieval tool-use distribution across training variants (*Other*: invalid tool labels). Naive-GRPO and GRPO $_{\Phi_{avg}}$ default to the generative retriever (84% and 85% of calls), as does Φ LeTS (74%) despite action-level credit; Φ MT-GRPO under-uses it (35%). PACE-GRPO prefers the cheaper cosine retriever, using the generative one only when needed (44%).

In addition, we carry out experiments considering our state potentials $\Phi(s_k)$ as an intrinsic reward for intermediate actions a_k , i.e., $\Phi(s_k) = r_k$. Then, we use the action-level reward normalisation proposed for MT-GRPO (Wei et al., 2025b; Xie et al., 2026) and the action-level advantage re-weighting proposed in LeTS (Zhang et al., 2025) to derive two additional variants that we call Φ MT-GRPO and Φ LeTS. We also include a Zero-shot baseline using PACE-GRPO’s prompt. Aside from accuracy (Table 2), we also show the resulting tool-use distributions in Figure 4.

All trained variants outperform the single-tool GRPO-R1 baseline, confirming that access to a second retrieval tool helps. However, without action-level feedback, Naive-GRPO converges to a degenerate policy that invokes the more expensive generative retriever on nearly every step, foregoing any benefit from the cheaper cosine similarity tool. GRPO $_{\Phi_{avg}}$ gains the most over Naive-GRPO by exploiting the aggregated potential signal. However, it also learns a policy that calls the generative retriever most of the time (85%). Averaging the potential over a rollout preserves information about the quality of the trajectory as a whole, but not about which individual call was necessary.

By assigning credit to individual actions, PACE-GRPO learns to reserve the generative retriever for sub-questions that require deeper semantic analysis. The Zero-shot baseline shows that this policy does not emerge from the prompt or tool interface alone: it performs worst (.43 Jdg), uses only 1.75 retrieval actions per question, and still produces invalid tool labels in 2% of calls. Nor are the state potentials

sufficient by themselves. Despite using the same potential signal, Φ MT-GRPO and Φ LeTS remain within .02 Jdg of Naive-GRPO and fail in opposite directions. Φ LeTS over-rewards repeated generative calls, which reliably obtain high retriever scores, leading to 74% generative use. In contrast, Φ MT-GRPO normalises intrinsic rewards across non-comparable steps, yielding a noisy signal and only 35% generative use. The key distinction is therefore not the potential itself, but how credit is assigned.

Compared with GRPO $_{\Phi_{avg}}$, the strongest ablation, PACE-GRPO gives up only .01 Jdg while halving training-time generative retrieval calls (30k vs. 60k). Averaging the potential over a rollout captures overall trajectory quality but cannot identify which individual calls were unnecessary. As a result, GRPO $_{\Phi_{avg}}$ reaches similar accuracy by invoking the generative retriever on 85% of steps. Training curves are shown in Figure 5, Appendix D. Examples of queries generated by PACE-GRPO for each of the retrieval tools are in Appendix E.

6 Conclusion

We presented PACE-GRPO, a variant of GRPO for retrieval-augmented question answering that addresses two limitations of standard outcome-only RL: reward sparsity and uniform credit assignment across all tokens in a rollout. PACE-GRPO computes a state potential from three inexpensive retrieval-environment signals (retrieval utility, silver passage coverage, and within-rollout passage diversity) and uses differences in consecutive potentials to provide action-level feedback without any gold intermediate annotations, teacher model, or additional forward passes. We also framed retrieval as a tool-selection problem, training the agent to choose between a fast similarity-based retriever and a more expensive generative retriever supervised entirely by utility gains from the environment.

Experiments on six multi-hop QA benchmarks show that PACE-GRPO consistently outperforms outcome-only GRPO and matches or exceeds more heavily supervised baselines, while learning a balanced tool-use policy that reserves the expensive retriever for sub-questions that genuinely require deeper semantic matching. These results suggest that rich, cheap signals already available in a retrieval environment can substitute for costly teacher supervision, opening a practical path toward efficient and accurate retrieval-augmented reasoning.

Limitations

PACE-GRPO derives its credit signal from the retrieval environment, and is therefore limited by the quality and coverage of that environment. In particular, our coverage potential relies on silver passages constructed offline from question–answer pairs rather than gold reasoning chains; these sets may be noisy, incomplete, or biased toward passages mentioning the final answer.

Our experiments also instantiate the tool space with only two retrievers, cosine similarity and generative, leaving open how the method scales to richer tool inventories such as web search, structured knowledge-base queries, or domain-specific retrievers. More broadly, the state potential combines heuristic retrieval utility, coverage, and diversity signals with fixed weights; these may require tuning across corpora and can over-reward novelty when repeated evidence is actually useful.

Finally, our evaluation focuses on multi-hop QA with short answers and Wikipedia-style evidence, and future work should test whether the same action-level credit signals transfer to long-form synthesis, specialised domains, and settings where answer-based silver evidence is unavailable.

Acknowledgments

We thank the anonymous reviewers and area chair for their constructive feedback. We are grateful to Jie He, Litu Ou, and Chuan Meng for facilitating the pre-processed versions of Wikipedia and BrowseCompPlus corpora. The authors gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council (grant EP/W002876/1). This work was also supported by the Edinburgh International Data Facility (EIDF) and the Data-Driven Innovation Programme at the University of Edinburgh. Access to EIDF was facilitated through the University of Edinburgh’s Generative AI Laboratory GAIL Fellow scheme.

References

Mohammad Hossein Amani, Aryo Lotfi, Nicolas Baldwin, Samy Bengio, Mehrdad Farajtabar, Emmanuel Abbe, and Robert West. 2026. [RL for reasoning by adaptively revealing rationales](#). In *The Fourteenth International Conference on Learning Representations*.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. [RQ-RAG: Learning to refine queries for retrieval aug-](#)

[mented generation](#). In *First Conference on Language Modeling*.

- Mingyang Chen, Linzhuang Sun, Tianpeng Li, Sun Haoze, Zhou Yijie, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. 2025. [Research: Learning to reason with search for LLMs via reinforcement learning](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Sahel Sharifymoghaddam, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Hosna Oyarhoseini, Nandan Thakur, Crystina Zhang, Luyu Gao, and 2 others. 2026. [Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent](#).
- Debrup Das, Sam O’Nuallain, and Razieh Rahimi. 2025. [RaDeR: Reasoning-aware dense retrieval models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19970–19997, Suzhou, China. Association for Computational Linguistics.
- DeepSeek-AI. 2025. [DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning](#). *Preprint*, arXiv:2501.12948.
- Gemma Team, Aishwarya Kamath, Johan Ferret, and et al. 2025. [Gemma 3 technical report](#). *Preprint*, arXiv:2503.19786.
- Anna Goldie, Azalia Mirhoseini, Hao Zhou, Irene Cai, and Christopher D Manning. 2025. [Synthetic data generation and multi-step reinforcement learning for reasoning and tool use](#). In *Second Conference on Language Modeling*.
- Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. [HippoRAG: Neurobiologically inspired long-term memory for large language models](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Jie He, Victor Gutiérrez-Basulto, and Jeff Z. Pan. 2025. [From sufficiency to reflection: Reinforcement-guided thinking quality in retrieval-augmented reasoning for llms](#). *Preprint*, arXiv:2507.22716.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. [Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Jian Hu, Xibin Wu, Wei Shen, Jason Klein Liu, Zilin Zhu, Weixun Wang, Songlin Jiang, Haoran Wang, Hao Chen, Bin Chen, Weikai Fang, Xianyu, Yu Cao, Haotian Xu, and Yiming Liu. 2025. [Openrlhf: An easy-to-use, scalable and high-performance rlhf framework](#). *Preprint*, arXiv:2405.11143.

- Abhinav Java, Srivathsan Koundinyan, Nagarajan Natarajan, and Amit Sharma. 2026. [FrugalRAG: Less is more in RL finetuning for multi-hop question answering](#). In *The Fourteenth International Conference on Learning Representations*.
- Yuelu Ji, Rui Meng, Zhuochun Li, and Daqing He. 2025. Curriculum guided reinforcement learning for efficient multi-hop retrieval augmented generation. *arXiv preprint arXiv:2505.17391*.
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025a. [Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning](#). In *Second Conference on Language Modeling*.
- Jiajie Jin, Yutao Zhu, Zhicheng Dou, Guanting Dong, Xinyu Yang, Chenghao Zhang, Tong Zhao, Zhao Yang, and Ji-Rong Wen. 2025b. [Flashrag: A modular toolkit for efficient retrieval-augmented generation research](#). In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 737–740, New York, NY, USA. Association for Computing Machinery.
- Tianjian Li, Yiming Zhang, Ping Yu, Swarnadeep Saha, Daniel Khashabi, Jason E Weston, Jack Lanchantin, and Tianlu Wang. 2026. [Jointly reinforcing diversity and quality in language model generations](#).
- Qi Luo, Xiaonan Li, Yuxin Wang, Tingshuo Fan, Yuan Li, Xinchu Chen, and Xipeng Qiu. 2025. MARAG-R1: Beyond single retriever via reinforcement-learned multi-tool agentic retrieval. *arXiv preprint arXiv:2510.27569*.
- Chuan Meng, Litu Ou, Sean MacAvaney, and Jeff Dalton. 2026. [Revisiting text ranking in deep research](#). In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '26, page 3006–3016, New York, NY, USA. Association for Computing Machinery.
- Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, page 278–287, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. [Document ranking with a pre-trained sequence-to-sequence model](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online. Association for Computational Linguistics.
- Laura Perez-Beltrachini and Mirella Lapata. 2025. [Uncertainty quantification in retrieval augmented question answering](#). *Transactions on Machine Learning Research*.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. [Measuring and narrowing the compositionality gap in language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.
- Hongjin Qian and Zheng Liu. 2025. [Scent of knowledge: Optimizing search-enhanced reasoning with information foraging](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. [Toolformer: Language models can teach themselves to use tools](#). In *Advances in Neural Information Processing Systems*, volume 36.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#). *CoRR*, abs/1707.06347.
- Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muenighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen tau Yih, Pang Wei Koh, and Luke Zettlemoyer. 2025. [ReasonIR: Training retrievers for reasoning tasks](#). In *Second Conference on Language Modeling*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *CoRR*, abs/2402.03300.
- Yaorui Shi, Sihang Li, Chang Wu, Zhiyuan Liu, Junfeng Fang, Hengxing Cai, An Zhang, and Xiang Wang. 2025. [Search and refine during think: Facilitating knowledge refinement for improved retrieval-augmented reasoning](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- 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](#). *Preprint*, arXiv:2503.05592.
- Kai Sun, Yifan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2024. [Head-to-tail: How knowledgeable are large language models \(LLMs\)? A.K.A. will LLMs replace knowledge graphs?](#) In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 311–325, Mexico City, Mexico. Association for Computational Linguistics.
- Hongze Tan, Zihan Wang, Jianfei Pan, Jinghao Lin, Hao Wang, Yifan Wu, Tao Chen, Zhihang Zheng, Zhihao Tang, and Haihua Yang. 2025. *GTPO*

- and GRPO-S: Token and sequence-level reward shaping with policy entropy. *arXiv preprint arXiv:2508.04349*.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. [MuSiQue: Multihop 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.
- Liang Wang, Haonan Chen, Nan Yang, Xiaolong Huang, Zhicheng Dou, and Furu Wei. 2025. [Chain-of-retrieval augmented generation](#). In *The Thirtieth Annual Conference on Neural Information Processing Systems*.
- Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. [Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, Bangkok, Thailand. Association for Computational Linguistics.
- Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeff Han, Isabella Fulford, Hyung Won Chung, Alexandre Passos, William Fedus, and Amelia Glaese. 2025a. [Browsecomp: A simple yet challenging benchmark for browsing agents](#). *ArXiv*, abs/2504.12516.
- Quan Wei, Siliang Zeng, Chenliang Li, William Brown, Oana Frunza, Wei Deng, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. 2025b. [Reinforcing multi-turn reasoning in LLM agents via turn-level reward design and credit assignment](#). In *First Workshop on Multi-Turn Interactions in Large Language Models*.
- Muning Wen, Ziyu Wan, Jun Wang, Weinan Zhang, and Ying Wen. 2024. [Reinforcing LLM agents via policy optimization with action decomposition](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. [C-pack: Packaged resources to advance general chinese embedding](#). *Preprint*, arXiv:2309.07597.
- Yutao Xie, Nathaniel Thomas, Nicklas Hansen, Yang Fu, Li Erran Li, and Xiaolong Wang. 2026. [TIPS: Turn-level information-potential reward shaping for search-augmented LLMs](#). In *The Fourteenth International Conference on Learning Representations*.
- Ran Xu, Wenqi Shi, Yuchen Zhuang, Yue Yu, Joyce C. Ho, Haoyu Wang, and Carl Yang. 2025. [Collab-RAG: Boosting retrieval-augmented generation for complex question answering via white-box and black-box LLM collaboration](#). In *Second Conference on Language Modeling*.
- 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](#). *Preprint*, arXiv:2505.09388.
- Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, and 8 others. 2024. [Crag - comprehensive rag benchmark](#). In *Proceedings of the 38th International Conference on Neural Information Processing Systems, NIPS '24*, Red Hook, NY, USA. Curran Associates Inc.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. 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*.
- Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael Tomz, Christopher D Manning, and Weiyan Shi. 2026. [Verbalized sampling: How to mitigate mode collapse and unlock LLM diversity](#).
- Qi Zhang, Shouqing Yang, Lirong Gao, Hao Chen, Xiaomeng Hu, Jinglei Chen, Jiexiang Wang, Sheng Guo, Bo Zheng, Haobo Wang, and Junbo Zhao. 2025. [LeTS: Learning to think-and-search via process-and-outcome reward hybridization](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5109–5122, Suzhou, China. Association for Computational Linguistics.

	Train	Dev	Total
# questions (MuSiQue-Ans)	19,938	2,417	22,355
Hop range	2–4	2–4	—
Supporting paragraphs / question	2–4	2–4	—

Table 3: Statistics for **MuSiQue-Ans** (Trivedi et al., 2022). We train on a sample of 9,000 questions and test on a sample from the provided development split following (Song et al., 2025; Chen et al., 2025; He et al., 2025); the full MuSiQue-Ans dataset additionally contains 2,459 test questions, for 24,814 answerable questions in total. Questions are constructed bottom-up from connected single-hop questions and filtered to reduce shortcut/disconnected reasoning.

A Datasets

MuSiQue (Trivedi et al., 2022) addresses a fundamental limitation of prior multi-hop benchmarks: their susceptibility to shortcut reasoning. Rather than collecting questions directly, MuSiQue constructs questions bottom-up by composing connected single-hop questions, where each hop is designed to depend on the output of a preceding hop. Its construction pipeline filters candidate compositions using the *MuSiQue condition*, which formalises connected reasoning by requiring that sub-questions cannot be answered independently of their required predecessor answers. Empirically, MuSiQue-Ans is substantially less susceptible to disconnected reasoning: the authors report that a single-hop model incurs an approximately 30-point drop in F1. The dataset contains 24,814 answerable questions, commonly described as 25K, spanning 2–4 hops, drawn from five seed QA datasets: SQuAD, Natural Questions, MLQA, T-REx, and Zero Shot RE. An extended variant, MuSiQue-Full, pairs each answerable question with a corresponding unanswerable contrast instance, created by making one constituent single-hop question unanswerable. We train on MuSiQue-Ans and report in-domain evaluation on its development split.

HotpotQA (Yang et al., 2018) is a large-scale multi-hop QA benchmark containing 112,779 Wikipedia-based question–answer pairs that require reasoning over multiple supporting documents. Questions are collected in two broad categories: *bridge* questions, where an intermediate entity connects two documents and must be inferred en route to the answer, and *comparison* questions, which require comparing attributes of two entities across documents. A distinctive feature

	Train	Dev	Total
# questions	90,564	7,405	97,969
Bridge questions	$\approx 81\%$	$\approx 81\%$	—
Comparison questions	$\approx 19\%$	$\approx 19\%$	—
Supporting docs / question	2	2	—
Supporting facts	sentence-level annotations		

Table 4: Statistics for **HotpotQA** (Yang et al., 2018). The test set is held out; dev figures are for the distractor setting. Bridge questions require chaining two documents via an intermediate entity; comparison questions require comparing attributes across two entities.

is the provision of sentence-level supporting-fact annotations, enabling both strong supervision for evidence selection and fine-grained evaluation of whether the model has identified the sentences supporting its answer. The dataset is evaluated in two settings: a *distractor* setting in which each question is paired with 10 candidate paragraphs, consisting of 2 gold paragraphs and 8 TF-IDF distractors, and a *full-wiki* setting in which systems must retrieve supporting evidence from the first paragraphs of all Wikipedia articles. We evaluate in the *full-wiki* setting assuming no training data beyond the gold answer.

2WikiMultiHopQA (Ho et al., 2020) is a large-scale multi-hop QA benchmark that combines structured and unstructured data to enforce multi-step reasoning. Each question is constructed from the intersection of Wikipedia text and Wiki-data knowledge-graph triples, using predefined templates over structured (subject, relation, object) facts and verifying the resulting questions against Wikipedia passages. A distinctive feature is the provision of explicit *evidence information*: structured triples that encode the reasoning path for each question, enabling fine-grained evaluation of reasoning steps beyond answer correctness alone.

The dataset covers four question types: *comparison* questions, which compare properties of two or more entities; *inference* questions, which are derived from logical rules in the knowledge graph (e.g., $father(a, b) \wedge father(b, c) \Rightarrow grandfather(a, c)$); *compositional* questions, which chain two triples through a bridge entity when no direct inference relation is available; and *bridge-comparison* questions, which combine bridge-entity reasoning with a subsequent comparison. Compositional questions are the most frequent type and inference questions the least frequent.

	Train	Dev	Test
# questions	167,454	12,576	12,576
Hops		two hops	
Evidence paths provided		✓	
Supporting facts provided		✓	
Question types	comparison, compositional inference, bridge-comparison		

Table 5: Statistics for **2WikiMultiHopQA** (Ho et al., 2020). The training set combines the official train-medium and train-hard splits. Questions are constructed from two-step reasoning patterns over Wikidata and Wikipedia. Evidence information provides explicit reasoning paths, while supporting facts identify the relevant Wikipedia sentences.

Property	Value
# questions (evaluation only)	125
Hops	2
Construction	Manual
Design criterion	incorrect search-engine snippet
Evidence source	Wikipedia
Template-based	No (questions are varied)

Table 6: Statistics for **Bamboogle** (Press et al., 2023). Bamboogle is a small, manually constructed evaluation benchmark with no training or development split. Questions are retained when a popular search engine returns an incorrect featured snippet, while both supporting facts can be found in Wikipedia.

Bamboogle (Press et al., 2023) is a small, manually constructed dataset of 125 two-hop compositional questions. The questions were written by the authors from Wikipedia articles and retained only if querying the full question in a popular internet search engine produced an incorrect featured snippet. This criterion provides evidence that the questions do not already appear on the web, and hence are unlikely to occur verbatim in LM pre-training data, while both supporting pieces of evidence remain available in Wikipedia. The dataset is therefore designed to test whether systems can compose two retrievable facts rather than rely on surface-level single-hop retrieval of the final answer.

Unlike larger automatically generated multi-hop datasets such as 2WikiMultiHopQA and MuSiQue, Bamboogle questions are deliberately varied: they cover different question types and domains and are not generated from a small set of templates. Because of its small size and manual construction, Bamboogle is used as an evaluation-only benchmark rather than as a source of training data.

Property	Value
# questions (evaluation only)	2,706
Construction	Manual
Evidence source	Web
Template-based	Mix of user and template-based
Question types	simple, simple-condition multi-hop, comparison, set aggregation, false-premise post-processing-heavy

Table 7: Statistics for **CRAG** (Yang et al., 2024). Questions are created from human-authored templates and sampled entities with different popularity as well as human-written questions reflecting information that users may search for.

For HotpotQA, 2WikiMultiHopQA, and MuSiQue, we adopt the test splits released by (Song et al., 2025) and further used in previous work (He et al., 2025; Chen et al., 2025), each containing 500 examples. For Bamboogle, we use the full set of 125 test examples.

CRAG (Yang et al., 2024) is a rich benchmark for the evaluation of RAG systems beyond Wikipedia composed of natural user-like questions. It contains 2,706 QA pairs (public subset) covering diverse domains (*Finance, Sports, Music, Movie, and Open domain*), eight question types (see Table 7), entities with varying popularity (popular to long-tail), and questions with different temporal dynamics (i.e., from facts that change in seconds to static ones). QA pairs are collected in two different ways. One is from human authored templates which are populated with entities from a custom KG. The other is by asking annotators to write queries that humans would ask and then use these to search Web content and produce the final QA pairs. Each QA pair is associated with up to 50 Web pages gathered using a search AP; the estimated recall is 74% for KG QA pairs and 93% for Web ones.

For our evaluation we only include those question types that involve search over multiple documents, i.e., multi-hop, comparison, simple-condition, and aggregation. We only include questions with static or slow-changing temporal dynamics to avoid temporal inconsistencies in our evaluation. The split we use in our experiments contains 607 questions. We take the entire collection of Web pages in CRAG’s public subset as our corpus. We use Trafilatura to process these Web pages, i.e., the tool used to extract content from HTML in Chen et al. (2026), and segment the extracted textual con-

Property	Value
# questions (evaluation only)	830
Construction	Manual
Design criterion	>10 min to solve by humans unsolvable by large scale LLMs
Evidence source	Web
Template-based	No (human written from seeds)
Question types	multihop
Average hops per question	8.7

Table 8: Statistics for **BrowseCompPlus** (Chen et al., 2026). Trained annotators write complex descriptions of seeds (e.g., entity or event) as complex multi-hop fact seeking questions.

tent into passages following Meng et al. (2026).³

BrowseCompPlus (Chen et al., 2026) is a benchmark of 830 complex questions derived from BrowseComp (Wei et al., 2025a), designed to evaluate agents’ ability to browse the Web and locate difficult-to-find information. Annotators construct questions from seed entities or events by combining multiple descriptive clues into a complex target query. Questions are designed to require more than ten minutes for a human to solve and to be beyond the capabilities of frontier LLMs available at the time of creation. Solving them therefore requires persistent search, strategic question decomposition, and effective tool use.

BrowseCompPlus augments these questions with verified evidence documents. Candidate URLs are collected semi-automatically using OpenAI o3 and subsequently checked by human annotators, who identify evidence spans and verify that they are jointly sufficient to answer each question. The resulting corpus contains 100,195 documents; each question is associated, on average, with 6.1 evidence documents, 2.6 documents containing the final answer, and 76.28 distractors. We use the pre-processed corpus from Meng et al. (2026).

B Implementation Details

Silver Passage Sets We assume a realistic retrieval setting in which the external document database is Wikipedia and no gold passages are provided (He et al., 2025; Chen et al., 2025). Nevertheless, intermediate signals about how close a reasoning step is to a correct answer have been shown to be beneficial (Qian and Liu, 2025; Xie et al., 2026). Inspired by work on offline corpus

Method	Average	
	CEM	Jdg
Vanilla RAG top-5	.08	.16
Vanilla RAG top-20	.11	.21
P^{silver} RAG	.31	.46

Table 9: Quality of silver passage sets P^{silver} on a 500-instance sample of the training set. Retrieval with silver passages substantially outperforms both Vanilla RAG top-5 and top-20, confirming that (x, y) -guided retrieval followed by re-ranking yields a useful approximation of gold evidence. CEM = Cover Exact Match; Jdg = LLM-judge accuracy; ‘-’ = not applicable (silver passages are not available at inference time).

indexing via entity graphs (Gutierrez et al., 2024), we approximate such signals by constructing a set of *silver* passages P^{silver} for each training instance without any gold annotation.

Concretely, given training pairs $\mathcal{D} = \{(x^i, y^i)\}$, we form a retrieval query $\bar{q}^i$ by concatenating question x^i and gold answer y^i , use it to retrieve 100 candidate passages with the lexical retriever, and re-rank them with the generative re-ranker, retaining the top 20 as P_i^{silver} . Intuitively, a query that includes the answer string biases retrieval towards passages covering both the question entities and the answer, which the re-ranker then orders by topical relevance.

Table 9 validates this construction on a random sample of 500 training instances by comparing Vanilla RAG against retrieval with P^{silver} . Because standard RAG outputs tend to be verbose, we report Cover Exact Match (CEM), which checks whether the gold answer string is contained in the predicted answer, alongside Jdg.

MT-GRPO Advantage Computation We re-implement MT-GRPO (Wei et al., 2025b) as formulated in (Xie et al., 2026). MT-GRPO is originally defined for rollouts with a single tool call. The general case requires normalising each intermediate step of each rollout in the group; thus, it requires an exponential number of samples. Xie et al. (2026) propose to normalise steps across rollouts within the group, i.e., normalise step k across all rollouts. If a rollout is longer than others in the group, normalisation is not defined and thus there is no intermediate reward for those extra steps of the longer rollout. Formally, every token i belonging to action a_k is modulated by intermediate rewards as follows:

$$A_{j,k,i} \leftarrow (1 - \beta) A_j^{\text{GRPO}} + \beta \tilde{r}_{j,k}, \quad (11)$$

³<https://trafilatura.readthedocs.io>

where $\tilde{r}_{j,k} = (r_{j,k} - \mu_k) / \sigma_k$ and $r_{j,k}$ is the intermediate reward for action a_k in rollout j , μ_k and σ_k are the mean and standard deviation of the rewards for action k across rollouts in the GRPO group. Tokens in the final answer action a_K only receive the outcome reward A_j^{GRPO} . In our experiments we set β to 0.1.

LeTS Advantage Computation We re-implement LeTS following its description in (Zhang et al., 2025).⁴ LeTS also incorporates intermediate rewards $r_{j,k}$ for action a_k in rollout j into the advantage computation. However, the intermediate reward normalisation is carried out within each rollout j , i.e., $\tilde{r}_{j,k} = (r_{j,k} - \mu_j) / \sigma_j$, where μ_j and σ_j are the mean and variance of intermediate actions within rollout j (excluding the final answer action a_K). Then every token i belonging to action a_k is modulated by intermediate rewards as follows:

$$A_{j,k,i} \leftarrow (1 + \text{sgn}(A_j^{\text{GRPO}}) \beta \tilde{r}_{j,k}) A_j^{\text{GRPO}}; \quad (12)$$

we set $\beta = 0.1$ as used in (Zhang et al., 2025).

Training We use QWEN3-4B-INSTRUCT (Yang et al., 2025) as the LLM backbone for PACE-GRPO and all fine-tuned baselines, trained with the hyperparameters listed in Table 10, and the group filtering mechanism introduced by (He et al., 2025). We implement our models and baselines using the Open-RLHF framework (Hu et al., 2025).

We do not perform an extensive search over PACE-GRPO’s hyperparameters. For the combined state potentials, we follow a straightforward criterion: we set the weights of the individual state potentials—retrieval, coverage, and diversity (ω_u , ω_c , ω_d)—to yield a similar contribution in the combined Φ . Recall that each individual state potential ranges over the $[0, 1]$ interval.⁵ For α , the strength of the action-level signal, we follow Xie et al. (2026): we observed the magnitude of the potential differences and, based on this, chose α to be neither too small (which would render its effect negligible) nor too large (which would cause the model to deviate from the original goal—e.g., stop using the generative tool). The state potential weights are set to $\omega_u=0.2$, $\omega_c=0.1$, and $\omega_d=0.1$, with $\alpha=0.05$, $\lambda_\Phi=0.8$, and $\lambda_{expl}=0.1$. For TIRESRAG-R1 we follow the original training configuration (He et al.,

⁴No implementation of LeTS has been released.

⁵In principle, since the silver passage set contains 20 passages, coverage can reach 1.2, although we never observe this in practice.

2025), using QWEN3-8B in its FP8-quantised variant as the teacher model, which reduces memory footprint while maintaining accuracy. For IRCOT, we set the number of iterations to eight as used in the experimental setup in (Trivedi et al., 2023).

Retrieval score Φ_u (Equation 1) is computed as follows. The set of top- N passages P_k returned for question q_k by any retrieval tool is scored by a single designated "retrieval-scoring" model, which ensures that scores are consistent across retrieval tools. In our implementation, we use the model employed by the generative tool. However, this could be any model able to assess the utility of the retrieved passage set, for example, a utility ranker as in (Perez-Beltrachini and Lapata, 2025). The maximum value over the passages in P_k is taken as the retrieval score. In addition, we penalise the retrieval score in three ways:

1. When the tool call invokes a non-existent tool, a fixed value of 0.1 is used as the retrieval score, i.e., a very small value, as retrieval scores range over $[0, 1]$.
2. Calls to the generative tool receive only 90% of the retrieval score.
3. When the retrieval score of a call to the expensive generative tool is close to that of the cheaper cosine tool (i.e., they differ by at most 0.05 points), only 60% of the retrieval score is taken.

Retrieval For the four Wikipedia-based benchmarks, following Song et al. (2025) and He et al. (2025), we use the 2019 English Wikipedia dump as the external knowledge source. Wikipedia pages are chunked into 100-token passages. For CRAG and BrowseCompPlus, we use the provided corpus of Web documents and pre-process these into 250-token passages following (Chen et al., 2026; Meng et al., 2026). We use BGE-large-en-v1.5 (Xiao et al., 2023) as the cosine similarity retriever and monot5-large-msmarco (Nogueira et al., 2020) as the generative retriever. Both retrievers return the top-5 passages, following standard practice in retrieval-augmented QA (Song et al., 2025; He et al., 2025) over a Wikipedia passage database. The initial candidate pool for re-ranking is set to 50 passages during training and expanded to 100 at inference to balance training speed against retrieval quality. For CRAG and BrowseCompPlus the retrievers are set to return the top-2 passages;

Parameter	Value
Advantage estimator	GRPO
γ	1.0
β	0
KL loss	True
KL estimator	K3
KL loss coef	0.001
clip ϵ	0.2
Episodes	2
train batch size	108
max prompt length	8192
max response length	8192
actor optim LR	2×10^{-6}
rollout max model len	15000
G	5
rollout temperature	1
VLLM USE V1 (env)	0
GPUS per node	4

Table 10: Hyperparameter settings shared across all GRPO-based models. G = rollout group size; β = KL penalty coefficient; clip ϵ = PPO clipping threshold.

this gives a similar context length to that used for the Wikipedia-based QA tasks.

Evaluation We use FlashRAG (Jin et al., 2025b) for evaluation pipeline management. We report results of one run for all models. LLM-judge accuracy (Jdg) is computed with GEMMA3-27B-INSTRUCT (Gemma Team et al., 2025) with an evaluation prompt based on that of Sun et al. (2024) which was shown to perform well on short phrase question answering evaluation (Sun et al., 2024; Perez-Beltrachini and Lapata, 2025). The prompt is reproduced in Appendix C.

C System Prompts

We provide the prompt templates used in our experiments. We show (1) the LLM-judge prompt used to compute the Jdg accuracy metric; (2) the tool-adaptive prompt used by PACE-GRPO and its ablation variants (Zero-Shot, Naive-GRPO, GRPO _{Φ_{avg}} , Φ MT-GRPO, and Φ LeTS); and (3) the standard search-augmented QA prompt used by the remaining baselines.

Accuracy Evaluation Prompt (Jdg)

You need to check whether the prediction of a question-answering system to a question is correct. You should make the judgment based on a list of ground truth answers provided to you. Your response should be "correct" if the prediction matches the ground truth answers and "incorrect" otherwise.

Question: Who authored The Taming of the Shrew (published in 2002)?
Ground truth: ["William Shakespeare", "Roma Gill"]
Prediction: W Shakespeare
Correctness: correct

Question: Who authored The Taming of the Shrew (published in 2002)?
Ground truth: ["William Shakespeare", "Roma Gill"]
Prediction: Roma Gill and W Shakespeare
Correctness: correct

Question: Who authored The Taming of the Shrew (published in 2002)?
Ground truth: ["William Shakespeare", "Roma Gill"]
Prediction: Roma Shakespeare
Correctness: incorrect

Question: What country is Maharashtra Metro Rail Corporation Limited located in?
Ground truth: ["India"]
Prediction: Maharashtra
Correctness: incorrect

Question: What's the job of Song Kang-ho in Parasite (2019)?
Ground truth: ["actor"]
Prediction: He plays the role of Kim Ki-taek, the patriarch of the Kim family.
Correctness: correct

Question: Which era did Michael Oakeshott belong to?
Ground truth: ["20th-century philosophy"]
Prediction: 20th century."
Correctness: correct

Question: Edward Tise (known for Full Metal Jacket (1987)) is in what department?
Ground truth: ["sound department"]
Prediction: 2nd Infantry Division, United States Army
Correctness: incorrect

Question: What wine region is Finger Lakes AVA a part of?
Ground truth: ["New York wine"]
Prediction: Finger Lakes AVA
Correctness: incorrect

Question: question
Ground truth: gold answer
Prediction: generated answer
Correctness:

Tool Adaptive Prompt

You are a helpful assistant. Answer the given question. You must reason **clearly and completely** about all pieces of information required to arrive at the final answer. Enclose your reasoning inside `<think>` and `</think>` tags. If you are uncertain about an entity or fact, or if the question requires external knowledge, you may need to issue simple sub-questions for external search to gather information. In the end, the sequence of simple sub-questions, gathered information, and reasoning steps should lead to the final answer.

Simple sub-questions can be of the following types:

- **parallel** (gather independent pieces of information for sub-parts of the question),
 - **sequential** (re-write a preceding question based on a previously gathered piece of information),
 - **interpretation** (re-write a preceding question for each possible concrete entity to which ambiguous entity mentions in the history may refer), or
 - **clarification** (gather information about relevant entities mentioned in the history to make sure reasoning is linked to appropriate concrete entities).
- If a sub-question fails to gather the necessary information, reformulate it in one of the following ways:
- **paraphrase** (using different words),
 - **general** (making it more general or vague --drop some details),
 - **specific** (making it more specific --add relevant details from the history), or
 - **reflect** (making it completely different --reflect about exploring different information gathering paths).
- Re-think the entire information search strategy if you are unable to locate the necessary information. Consider different ways the question could be understood. Look for potential ambiguities in terminology, scope, or intent.

Choose one of the above sub-question types; then generate the corresponding sub-question enclosed within `<search>`your sub-question`</search>` tags. **Carefully think about the search strategy** for the sub-question. The search strategy can be one of the following:

- Shallow: information can be found with a cheap similarity search over the external resource
- Deep: information needs a deep search against the external resource.

Generate the strategy enclosed within `<strategy>` and `</strategy>` tags.

Next to each sub-question, a set of passages will be provided enclosed within `<information>` and `</information>` tags.

Please **do not repeat simple questions** you have already issued, as this is a waste of time. Please **do not use coreference** such as 'this' within simple questions, always use the entity mention as introduced before. Carefully reflect on the formulated simple questions and each piece of information within the set of provided passages. After you think you have gathered all the necessary information, before providing your final answer, rethink it to make sure the answer is exactly correct for the original question. Only after satisfying all the above, give your final short phrase answer inside `<answer>` and `</answer>` tags.

Output format:

```
<answer> your short phrase answer </answer>
or
<strategy> only one of: Shallow or Deep </strategy>
<search>your sub-question</search>.
```

Question: `question`

Standard Search Augmented QA Prompt

You are a helpful assistant. Answer the given question. You must reason **clearly and completely** inside `<think>` and `</think>` before providing any final answer. Always identify and verify all key entities (e.g., person names, locations, dates, awards) mentioned in the question. If you are uncertain about an entity or fact, or if the question requires external knowledge, you may use `<search>`your query`</search>`, and the top search results will be returned between `<information>` and `</information>`. Carefully read and reflect on each newly retrieved piece of information. You can search as many times as you want. When reasoning, you must ensure your reasoning path aligns strictly with the evidence. After reasoning, before providing your final answer, rethink it to make sure the answer is exactly correct for the original question. Use the most accurate span from the evidence when possible. Only after satisfying all the above, give the final answer inside `<answer>` and `</answer>`. For example, `<answer> Beijing </answer>`. After outputting the final answer in `<answer>` `</answer>`, You have one chance to reflect on the answer. If you choose not to reflect, nothing further needs to be done. Otherwise, you can then re-examine your thinking process, the information obtained, and even search for more information to verify your previous answer or correct the previous answer. Remember, after reflection ends, you should output the answer in `<answer>` `</answer>`.

Question: `question`

D Additional Results

Tool Usage Throughout Training Figure 5 shows the proportion of generative-retriever calls over training. Naive-GRPO, GRPO* Φ * avg, and Φ LeTS increasingly favour the expensive retriever and eventually use it for most retrieval steps. Φ MT-GRPO initially explores policies with high generative-retriever use, but later shifts towards the cheaper cosine retriever. In contrast, PACE-

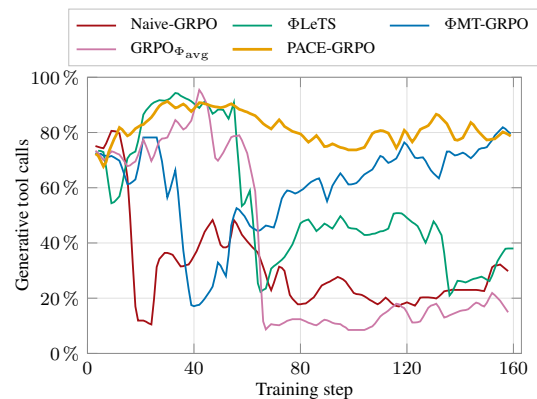


Figure 5: Proportion of generative tool calls throughout training. The yellow line corresponds to PACE-GRPO and blue to Φ MT-GRPO. Φ LeTS (green), GRPO $_{avg}(\Phi)$ (pink), and Naive-GRPO (red) degenerate to using the generative tool most of the time.

GRPO maintains substantially lower generative-retriever use throughout training. Its action-level credit signal identifies when individual expensive calls are beneficial, allowing it to learn a selective tool-use policy from early in training. PACE-GRPO therefore reduces the number of costly retrieval operations incurred during training while retaining strong downstream performance.

Ablation on Potential Components We carried out a sensitivity analysis of PACE-GRPO’s main hyperparameters by reducing them to zero one at a time. We report Jdg values and average number of steps in Table 13 for the Wikipedia-based datasets. When these components are disabled, PACE-GRPO’s performance drops. Interestingly, when the state coverage is removed, the state diversity signal prevails and search actions receive less reward leading to shorter trajectories (in terms of number of steps). Removing the state diversity also reduces performance, and trajectories are slightly longer. The process reward is also necessary for GRPO preferences among trajectories. The group diversity reward has the smallest impact and seems only beneficial on the in-domain dataset.

Full Results Table 11 and Table 12 show detailed results per metric and benchmark corresponding to Table 1 and Table 2 in the main section respectively.

E Examples

Table 15 shows example traces (i.e., sequences of tool calls, queries, and answers) for PACE-GRPO and Naive-GRPO. Naive-GRPO relies on the gen-

Method	MuSiQue			HotPotQA			2Wiki			Bamboogle			CRAG			BCPlus			Average			S
	EM	F1	Jdg	EM	F1	Jdg	EM	F1	Jdg	EM	F1	Jdg	EM	F1	Jdg	EM	F1	Jdg	EM	F1	Jdg	
Closed-Book	.02	.08	.07	.18	.24	.29	.26	.29	.31	.05	.13	.10	.17	.24	.29	.00	.04	.02	.11	.17	.18	—
Vanilla RAG	.06	.12	.13	.33	.43	.52	.07	.13	.39	.17	.27	.34	.24	.31	.43	.02	.05	.05	.15	.22	.31	—
IRCoT	.07	.13	.23	.31	.41	.60	.22	.30	.48	.29	.37	.49	.18	.24	.41	.01	.02	.04	.18	.24	.37	—
Zero-Shot	.10	.18	.18	.28	.37	.42	.29	.36	.38	.41	.51	.52	.29	.36	.45	.02	.05	.05	.23	.31	.33	1.03
GRPO-R1	.27	.38	.41	.49	.62	.73	.63	.72	.79	.50	.61	.62	.34	.41	.55	.05	.09	.07	.38	.47	.53	4.50
TIRESRAG-R1	.21	.32	.36	.48	.62	.73	.56	.67	.76	.49	.61	.63	.35	.42	.54	.07	.07	.09	.35	.45	.52	3.60
MT-GRPO	.24	.36	.40	.47	.61	.71	.64	.72	.77	.50	.63	.66	.35	.41	.55	.07	.11	.10	.38	.47	.53	5.51
LeTS	.27	.40	.44	.51	.65	.75	.63	.71	.79	.53	.66	.69	.35	.42	.56	.07	.10	.11	.39	.49	.56	4.31
PACE-GRPO	.27	.40	.45	.50	.64	.76	.63	.71	.81	.53	.66	.68	.34	.41	.59	.20	.22	.23	.41	.51	.59	5.00

Table 11: Results across six multi-hop QA benchmarks (QWEN3-4B-INSTRUCT backbone, all systems retrained under an identical setup). All datasets except for MuSiQue are out-of-domain; PACE-GRPO outperforms all baselines on every dataset and metric, matching or exceeding more heavily supervised methods while relying solely on inexpensive retrieval-environment signals. Best result per column is in **bold**. Jdg is a shorthand for LLM-judge accuracy; Steps are the average number of retrieval actions per question; ‘—’ means not applicable. **BCPlus** abbreviates BrowseCompPlus.

Method	MuSiQue			HotPotQA			2Wiki			Bamboogle			CRAG			BCPlus			Average			S
	EM	F1	Jdg	EM	F1	Jdg	EM	F1	Jdg	EM	F1	Jdg	EM	F1	Jdg	EM	F1	Jdg	EM	F1	Jdg	
Zero-shot	.11	.21	.22	.32	.45	.60	.35	.47	.57	.42	.51	.54	.31	.38	.52	.07	.10	.11	.26	.35	.43	1.75
Naive-GRPO	.25	.38	.42	.48	.63	.73	.60	.69	.77	<u>.54</u>	.65	.67	.31	.40	.56	.13	.20	.19	.39	.49	.56	2.49
GRPO _{Φ_{avg}}	.28	.41	.44	.49	.63	.75	.65	.74	.84	.60	.72	.74	.34	.41	.59	.17	.23	.24	.42	.52	.60	3.67
Φ MT-GRPO	.26	.37	.41	.48	.63	.74	.60	.68	.77	<u>.54</u>	<u>.66</u>	.66	.32	.39	.55	.12	.17	.17	.39	.48	.55	3.41
Φ LeTS	.25	.38	.44	.45	.60	.75	.58	.67	.78	.51	.63	.67	.30	.39	.57	.14	.18	.21	.37	.48	.57	3.22
PACE-GRPO	<u>.27</u>	<u>.40</u>	.45	.50	.64	.76	<u>.63</u>	<u>.71</u>	<u>.81</u>	.53	<u>.66</u>	<u>.68</u>	.34	.41	.59	.20	<u>.22</u>	<u>.23</u>	<u>.41</u>	<u>.51</u>	<u>.59</u>	5.00

Table 12: Results across six multi-hop QA benchmarks (QWEN3-4B-INSTRUCT backbone, all systems retrained under an identical setup). All datasets except for MuSiQue are out-of-domain; GRPO _{Φ_{avg}} is the strongest method at the expense of using the generative tool 85% of the time. PACE-GRPO achieves close or better performance using only the expensive tool 44% of the time. Best result per column is in **bold** and second best underlined. Jdg is a shorthand for LLM-judge accuracy; Steps are the average number of retrieval actions per question; ‘—’ means not applicable. **BCPlus** abbreviates BrowseCompPlus.

erative retriever even when questions can be solved with cheaper cosine retrieval. In contrast, PACE-GRPO invokes generative retrieval when it cannot resolve the question using evidence returned by the cheaper retriever.

Method	MuSiQue	HotPotQA	2Wiki	Bamboogle	Avg.	Steps
PACE-GRPO	.45	.76	.81	.68	.67	3.27
- state coverage ($\omega_c=0$)	.23	.60	.55	.50	.47	1.54
- state diversity ($\omega_d=0$)	.43	.72	.78	.68	.65	3.43
- group diversity ($\lambda_{expl}=0$)	.43	.74	.80	.70	.67	3.42
- process reward ($\lambda_\Phi = 0$)	.42	.70	.75	.63	.63	3.97

Table 13: Ablation of PACE-GRPO hyperparameters. Jdg refers to LLM-judge accuracy; Steps are the average number of retrieval actions per question.

Lexical Query	MuSiQue	Deep Query
Who played Thelma in the movie Thelma and Louise?	Who is Jack in the context of a covert mission involving 14th-century musical notation?	
Who is the author of the book titled "Queen of the Elephants"?	Who is Pao Sarasin and where was she born?	
Who performed the album Something Like a Bird?	How many times did the plague occur in Venice?	
Where was Levi Casey born?	Official record of marriage between Frank Sinatra and Lauren Bacall?	
Where is the Superior Drill Company located?	When did John Everett Millais die?	
Which record label is Christopher Cross associated with?	What significant historical event or period is commonly associated with the "disgrace of the Near East" in historical scholarship?	
When did the Real Housewives of Atlanta series start?	Who is the top scorer in the 2023-2024 EPL season according to ESPN or Sky Sports?	
Who gives out the Adam Smith Prize?		
What did Henry Moseley (physicist) participate in?		
Lexical Query	Bamboogle	Deep Query
birth date of Mykola Leontovych	What novel is the musical Cats based on?	
Which club did Kaká play for in 2012?	Official result of the women's indoor volleyball final at the 2012 Summer Olympics in London	
when was the Akashi Kaikyo Bridge opened	What is the longest running show in Broadway history? Who was the father of Charles Tupper? how many grams of protein in one large egg yolk Official Nobel Prize website: Walther Bothe biography What is the most commonly cited real-life inspiration for the character Russ Hanneman in Silicon Valley?	
What country is the Spring Temple Buddha located in?		
When was the KVLV-TV mast completed?		
When was New Delhi established?		
In what year was Colombia founded?		
Lexical Query	HotpotQA	Deep Query
What is the architect of the Rijksmuseum?	What electoral division in New South Wales was established in 1906 and includes the suburb of Alexandria?	
Who designed the Royal Festival Hall?	What was the name of the fifth hurricane of the 1966 Atlantic hurricane season?	
Publication date of the first book in Stephen King's Dark Tower series	When was the Big East Conference created?	
When was Walter Dornberger born?	Most recent annual passenger traffic for Palm Springs International Airport (as of 2023 or 2024)	
What is the location of Estadio Victoria?	What was the original attraction at Disneyland that inspired the Matterhorn Bobsleds, and how is it connected to the Matterhorn mountain?	
Willsmere railway station in Kew, Melbourne		
Lexical Query	2WikiMultiHopQA	Deep Query
Where did Dean Parisot study?	What is the release date of the film Sofia's Last Ambulance?	
Who directed The Mystery Of Pine Creek Camp?	Biographical information about Sturla Gunnarsson director of Beowulf & Grendel	
Where was Stacy Peralta born?	Biography of Mashonda Tifere education details	
What is Tim Wheeler's nationality?	Who were the sisters of Augusta von Fersen?	
Who directed McVeagh of the South Seas?	Any notable individuals with surname Generelo?	
Where was Adalbert II, Margrave of Tuscany, born?	Is Tadbaba Maryam located in Sayint?	
What country is Edwin L. Marin from?	Death date of Andrew Marton, director of Men of the Fighting Lady	
How old was Ralph Ince when he died?	Where did Edward Fitzwilliam, the actor born in 1788, die?	

Table 14: Examples of lexical and deep semantic search queries issued by PACE-GRPO for the cosine similarity and generative retrievers, respectively.

PACE-GRPO	Naive-GRPO
What was Nintendo's limit on games per developer per year for the platform with an abbreviated name that had the game Sunman? A: five	
<tool>Cosine</tool><search>What platform is associated with the game Sunman?</search> <tool>Cosine</tool><search>What was Nintendo's limit on games per developer per year for the Nintendo Entertainment System (NES)?</search> <answer>five games per year</answer>	<tool>Generative</tool> <search>Which gaming platform had a game titled Sunman?</search> <tool>Generative</tool> <search>What was Nintendo's limit on games per developer per year for the Nintendo Entertainment System (NES)?</search> <answer>five</answer>
Why did Roncalli leave the place of death of the maker of The Gozzi Altarpiece? A: for the conclave in Rome	
<tool>Cosine</tool> <search>Who was the maker of The Gozzi Altarpiece?</search> <tool>Cosine</tool> <search>Where did Titian die?</search> <tool>Cosine</tool> <search>Who was Roncalli?</search> <tool>Cosine</tool> <search>Why did Pope John XXIII leave Venice?</search> <answer>to attend the papal conclave in Rome</answer>	<tool>Generative</tool> <search>Who created the Gozzi Altarpiece and where did the artist die?</search> <tool>Generative</tool> <search>Did Benozzo Gozzoli create an altarpiece known as the Gozzi Altarpiece?</search> <answer>There is no historical connection between Roncalli and the maker of The Gozzi Altarpiece, so the premise of the question is invalid.</answer>
What is the date of birth for the performer of Beat Hotel? A: 11 September 1962	
<tool>Cosine</tool> <search>Who is the performer of "Beat Hotel"? And what is their date of birth?</search> <tool>Cosine</tool> <search>What is Mauro Scocco's date of birth?</search> <tool>Generative</tool> <search>What is the date of birth of Mauro Scocco, the Swedish pop music artist who released the album Beat Hotel in 2003?</search> <answer>11 September 1962</answer>	<tool>Generative</tool> <search>Who performed the song "Beat Hotel"?</search> <tool>Generative</tool> <search>What is the date of birth of members of the band Biff Bang Pow!</search> <tool>Generative</tool> <search>What is the date of birth of Alan McGee, the founder of Biff Bang Pow!</search> <answer>29 September 1960</answer>

Table 15: Examples of the sequence of tool calls in PACE-GRPO and Naive-GRPO from the MuSiQue dataset.