LongTraceRL: Learning Long-Context Reasoning from Search Agent Trajectories with Rubric Rewards

Anonymous ACL submission

Abstract

Long-context reasoning remains a central challenge for large language models, which often fail to locate and integrate key information in extensive distracting content. Reinforcement learning with verifiable rewards (RLVR) has shown promise for this task, yet existing methods are limited by low-confusability distractors and sparse, outcome-only reward signals that cannot supervise intermediate reasoning steps. To address these issues, we introduce LONGTRACERL. For data construction, we generate multi-hop questions via knowledge graph random walks and leverage search agent trajectories to build *tiered distractors*: documents the agent read but did not cite (high confusability) and documents that appeared in search results but were never opened (low confusability), producing training contexts that are far more challenging than those built by random sampling or one-shot search. For reward design, we propose a *rubric reward* that uses the gold entities along each reasoning chain as fine-grained, entity-level process supervision. This rubric reward is applied only to responses with correct final answers (positive-only strategy), distinguishing the reasoning quality among correct responses and preventing reward hacking. Experiments on three reasoning LLMs (4B–30B) across five long-context benchmarks demonstrate that LONGTRACERL consistently outperforms strong baselines and encourages comprehensive, evidence-grounded reasoning.

1 Introduction

Long-context reasoning is a critical capability for large language models (LLMs), driving advances in both single-pass reasoning (Bai et al., 2025; Team, 2025a) and multi-turn autonomous agent systems (Yao et al., 2023; Shinn et al., 2023; Yang et al., 2024) by enabling models to extract key information across a global context, perform multi-hop inference, and stay coherent over extended text. Despite its importance, current LLMs still struggle with long-context understanding (Guan et al., 2026), especially in realistic scenarios full of distracting information. As the context length grows, they often exhibit typical failure patterns: giving hallucinated answers, relying on fragmented retrieval, or citing irrelevant passages. These limitations make long-text reasoning a major bottleneck for deploying reasoning-oriented models in real-world applications. Recently, reinforcement learning with verifiable rewards (RLVR) has proven effective for multiple tasks such as mathematical reasoning (DeepSeek-AI, 2025; Shao et al., 2024)

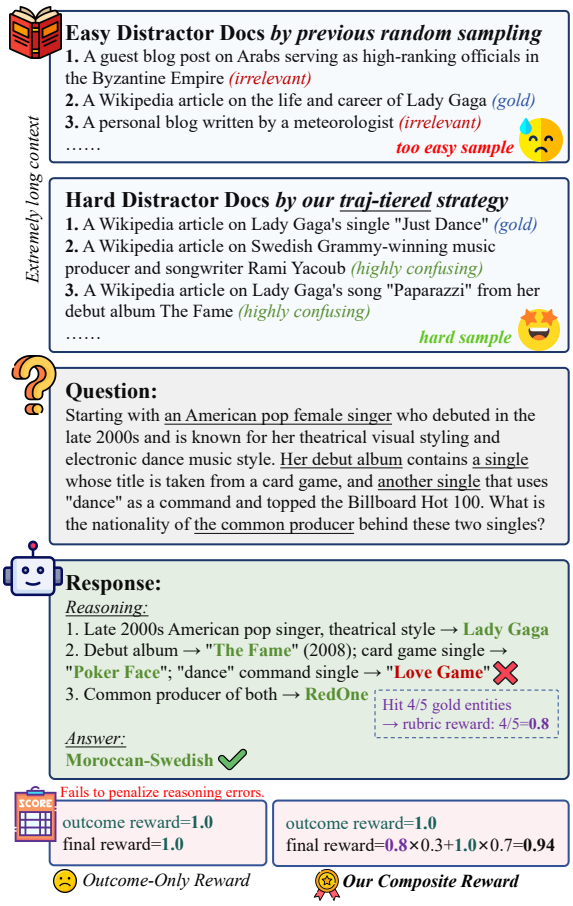


Figure 1: Comparison between prior long-context RL approaches based on easy distractors and outcome-only rewards, and our proposed LONGTRACERL.

and long-context question answering (Chen et al., 2026; Wang et al., 2025). However, current long-context RL methods have two key limitations. First, the quality of training data remains limited. Existing methods construct questions with few reasoning hops and shallow chains, with their distractors mostly sampled randomly from unrelated documents (Wang et al., 2025; Guan et al., 2026), lacking semantic relevance to the query and providing limited confusability as shown in Figure 1. Second, the reward signal is too sparse. Existing methods primarily rely on outcome-based rewards, providing optimization guidance solely based on the correctness of the final answer. When the input spans tens or even hundreds of thousands of tokens, such rewards become very sparse and can be noisy: the model may reach the correct answer through a wrong reasoning path by chance. For example, as shown in Figure 1, the model correctly answers "Moroccan-Swedish" while actually citing the wrong entity "Love Game" instead of "Just Dance" at an intermediate hop. Such coincidental successes satisfy the binary outcome reward but mask retrieval failures in intermediate steps.

To address these limitations, we propose **LongTraceRL**, which tackles both data construction and reward design. On the data side, inspired by Lu et al. (2025), we generate complex multi-hop questions with extremely long reasoning chains via knowledge graph random walks over the KILT Wikipedia snapshot (Petroni et al., 2021), and introduce a novel approach that constructs distractors based on real search trajectories from a search agent. Specifically, documents that the agent read but did not cite in the final response serve as high-confusability distractors (Tier-1), while documents that appeared in search results but were never opened serve as low-confusability distractors (Tier-2). Compared to random sampling or single direct search, these distractors are more relevant to the query, forcing the model to distinguish more carefully and reason more deeply. On the reward side, we design a rubric reward that uses the gold entities at each hop of the reasoning chain as fine-grained, entity-level process supervision. This reward is applied only to responses with correct final answers (positive-only strategy), helping distinguish the reasoning quality among correct responses and effectively preventing the model from gaming the reward by skipping intermediate reasoning steps and guessing the answer directly. Experiments on three reasoning LLMs (4B–30B) across five long-context benchmarks demonstrate that LongTraceRL consistently outperforms all baselines and encourages comprehensive, evidence-grounded reasoning, with Qwen3-4B achieving an average gain of 5.7 points over the base model and surpassing the strongest baseline by 2.5 points.

Our main contributions are as follows: (1) We propose a long-context training data construction method based on search agent trajectories, using tiered distractors to significantly improve the challenge and realism of the training data. (2) We design an entity-level rubric reward that provides finer-grained supervision on intermediate reasoning than the existing approaches for long-context reinforcement learning. (3) Through comprehensive experiments, we demonstrate the consistent improvements of LongTraceRL across multiple model families and scales.

2 Related Work

Long-Context Synthetic Data. Synthesizing long-context training data typically involves two design choices: how to construct questions and how to assemble the context. For question construction, some work (Wang et al., 2025; Li et al., 2024; Zhu et al., 2025a; Guan et al., 2026) reuses short-context multi-hop QA datasets such as MuSiQue (Trivedi et al., 2022) and HotpotQA (Yang et al., 2018), while others generate questions from scratch (Bai et al., 2024; Chen et al., 2025). More recent work uses structured knowledge to produce deeper reasoning chains, e.g., QwenLong-L1.5 (Shen et al., 2025) samples multi-hop paths from document-derived knowledge graphs, and DeepDive (Lu et al., 2025) performs random walks over the Wikipedia knowledge graph. For context assembly, some methods use a single long document (Bai et al., 2024; Chen et al., 2025, 2026), while others extend short-context QA datasets by adding distractor documents (Wang et al., 2025; Li et al., 2024; Zhu et al., 2025a; Guan et al., 2026). These distractors are usually sampled at random and are easy to filter out. NExtLong (Gao et al., 2025) improves this by using hard negative mining from dense retrieval, but its distractors are still based on embedding similarity rather than realistic search behavior, leaving a gap with practical retrieval scenarios.

Long-context Reinforcement Learning. While RLVR has proven effective on self-contained reasoning tasks such as mathematics (DeepSeek-AI,

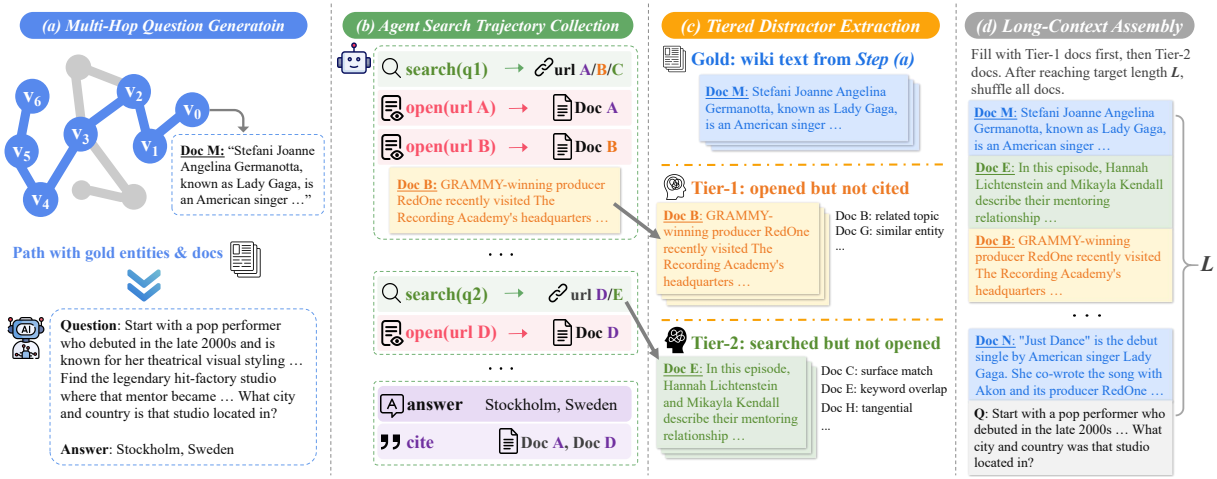


Figure 2: Overview of the LONGTRACERL training data construction pipeline.

2025; Shao et al., 2024), its adaptation to long-context scenarios remains limited, since outcome-based rewards supervise only the final answer and give no signal for intermediate reasoning over a large input. To address this, Chen et al. (2026) uses a chunk-level context reward based on F_β scores between predicted and gold document chunks, Guan et al. (2026) provides dense process-level supervision on evidence extraction quality via a co-evolving reward model, and LongR (Ping et al., 2026) measures relative information gain from retrieved documents under a frozen verifier. In parallel, fine-grained process rewards have also been explored in agentic RL: Zhang et al. (2026) decompose deep-search answers into citation-aware rubric items for verifiable reward, and Singh et al. (2025) shape the reward by classifying the cognitive behavior and utility of each tool call to reduce reward hacking. However, these methods either operate at the chunk, document, or tool-call level, or require an auxiliary LLM for evidence scoring at additional cost, leaving finer-grained, entity-level reasoning supervision unexplored.

3 Method

LONGTRACERL framework consists of two main components: (1) a data construction pipeline that synthesizes long-context training data with agent-derived distractors (§3.1), and (2) a reinforcement learning framework that combines outcome-based and process-based rewards (§3.2).

3.1 Data Construction Pipeline

We construct long-context training data through a four-step pipeline. The key idea is to leverage knowledge graph structure to generate multi-hop questions with verifiable reasoning chains, and then use agent search behavior to produce realistic distractors with high confusability. Figure 2 provides an overview of the entire pipeline.

3.1.1 Multi-Hop Question Generation

Inspired by Lu et al. (2025), we automatically generate multi-hop questions from the KILT Wikipedia snapshot (Petroni et al., 2021) through a two-stage process: knowledge graph random walk and question synthesis.

Knowledge Graph Random Walk. We perform controlled random walks over the Wikipedia hyperlink graph to collect multi-hop entity paths. Starting from a seed entity v_0, we walk $k(= 8)$ steps following hyperlinks to form a path $P = [v_0, v_1, \ldots, v_k]$. At each step, an LLM selects the next most relevant entity from up to five unvisited candidates. Between path collections, we insert periodic *mad walks* (a few random jumps) to diversify the explored graph regions.

Question Synthesis. Given a path P and the Wikipedia text of each entity, we prompt a powerful LLM (such as GPT-5.2) to generate a multi-hop question whose answer is a specific attribute of the last entity v_k. As shown in Figure 12, the prompt enforces several constraints: (i) the question must require step-by-step reasoning through all entities in the path, with no shortcuts; (ii) all identifying information (names, dates, locations) must be paraphrased so that the answer cannot be found by simple keyword matching; and (iii) the answer must be unique and exactly match the selected attribute. The LLM also outputs the list of

intermediate gold entities along the reasoning chain. For each question, we thus obtain the question text, the ground-truth answer, and the set of gold entities $\mathcal{E} = \{e_1, e_2, \ldots, e_k\}$ with their corresponding Wikipedia passages.

3.1.2 Agent Search Trajectory Collection

To construct realistic distractors, we leverage the behavioral trajectories of a search agent attempting to answer each generated question. We deploy an agent equipped with deep search capabilities, including issuing search queries (SEARCH), opening and reading retrieved documents (OPEN), and citing information in its response (CITE). We record the agent's complete search trajectory $\tau = [(a_1, d_1), (a_2, d_2), \ldots]$, where a_t denotes the action type and d_t the associated document.

Trajectory Filtering. To obtain correct reliable trajectories, we sample $K(= 5)$ independent trajectories per question and retain only those where the agent reaches the final correct answer. For each question, one of the correct trajectories is selected for subsequent distractor extraction, and the questions where all K attempts fail are discarded. This filtering ensures that the retained trajectories reflect real, meaningful, and goal-directed search behavior rather than random exploration or hallucination.

3.1.3 Tiered Distractor Extraction

From the recorded search trajectories, we divide the retrieved documents (excluding gold evidence passages) into two tiers: (1) **Tier-1 distractors** (high confusability): documents that the agent opened and read but did not cite in its final response, which are topically relevant and were initially deemed worth reading, making them strong distractors. (2) **Tier-2 distractors** (low confusability): documents that appeared in search results but were never opened, which are only superficially related to the query and are less likely to mislead a careful reader.

3.1.4 Long-Context Assembly

The final long-context input is assembled following a strategy named **`traj-tiered`** (short for trajectory-tiered). Starting from the gold passages, we first add Tier-1 distractors $\mathcal{D}_1$, which are more confusing and thus more valuable for training. If the context has not yet reached the target length L after exhausting all Tier-1 distractors, we continue to fill with Tier-2 distractors $\mathcal{D}_2$. This prioritization ensures that the model is exposed to as many challenging distractors as possible. All documents are then shuffled to prevent positional shortcuts.

3.2 RL with Rubric Reward

Since the data construction pipeline provides gold entities along each reasoning chain, we can leverage them as process supervision during reinforcement learning. We adopt Group Relative Policy Optimization (GRPO; Shao et al., 2024) as our RL algorithm and design a composite reward that combines an outcome reward for answer correctness with a rubric reward for reasoning quality.

Outcome Reward. The binary outcome reward $r_{\text{oc}} \in \{0, 1\}$ evaluates whether the model's final short answer is correct, as determined by an LLM judge.

Rubric Reward. The raw rubric score $\hat{r}_{\text{rb}}$ measures the recall of gold entities $\mathcal{E}$ in the model's response:

$$\hat{r}_{\text{rb}} = \frac{|\{e \in \mathcal{E} \mid e \text{ appears in the response}\}|}{|\mathcal{E}|} \quad (1)$$

A higher $\hat{r}_{\text{rb}}$ indicates that the model referenced more of the gold entities in reasoning.

Group-Level Rubric Normalization. In GRPO, each training question is answered by a group of G sampled responses. Since different questions involve different numbers of gold entities and varying difficulty levels, the raw rubric scores may span different ranges across questions. To ensure comparability, we normalize $\hat{r}_{\text{rb}}$ within each group by dividing by the group maximum when it is positive:

$$r_{\text{rb}} = \begin{cases} \dfrac{\hat{r}_{\text{rb}}}{\max_{j \in [G]} \hat{r}_{\text{rb}}^{(j)}}, & \text{if } \max_{j \in [G]} \hat{r}_{\text{rb}}^{(j)} > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

This rescales the rubric reward to $[0, 1]$ within each group, providing consistent process signals regardless of question difficulty.

Positive-Only Reward Combination. Since the rubric reward is based on entity recall, applying it to all responses risks *reward hacking*: the model could learn to enumerate entities mentioned in the retrieved passages rather than genuinely reasoning over them, inflating the rubric score without actually solving the problem. To prevent this, we adopt a **`positive-only`** strategy in which the rubric re-

ward is only granted to responses whose final answer is correct:

$$r = \begin{cases} (1-\alpha) \cdot r_{\text{oc}} + \alpha \cdot r_{\text{rb}}, & \text{if } r_{\text{oc}} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

In this formulation, the rubric reward serves to differentiate among correct responses, assigning higher scores to those that provide sound intermediate reasoning and lower scores to those that arrive at the right answer through shortcuts. Incorrect responses simply receive zero reward. The hyperparameter $\alpha \in [0, 1]$ controls the weight of process supervision: when $\alpha = 0$, the reward reduces to standard outcome-based GRPO; as α increases, the model receives stronger incentives to ground its reasoning in the relevant evidence passages.

4 Experiments

4.1 Setup

Models. We experiment with several reasoning-capable LLMs of different families and sizes to test generalizability: (1) **Qwen3-4B-Thinking-2507** (Team, 2025b), a 4B dense reasoning model; (2) **DeepSeek-R1-0528-Qwen3-8B** (DeepSeek-AI, 2025), a distilled dense model from the updated DeepSeek-R1-0528; (3) **Qwen3-30B-A3B-Thinking-2507** (Team, 2025b), a mixture-of-experts model with 30B total parameters and 3B active parameters. All ablation studies are conducted using Qwen3-4B-Thinking-2507.

Datasets. Our training set consists of 2,815 long-context QA examples constructed via the pipeline described in §3.1. Each example contains a eight-hop question, gold evidence passages from Wiki, and tiered distractors assembled to a target context length of 128K tokens. We compare our method against three existing long-context RL datasets, using the same RL algorithm and hyperparameters for a fair comparison: (1) **DocQA** (Wan et al., 2025): 1,591 QA examples on real documents covering math, logic and multi-hop reasoning, with context lengths ranging from 2K to 20K tokens. (2) **LoongRL** (Wang et al., 2025): 15,000 QA examples of 16K tokens, built by the KeyChain pipeline that pads short-context multi-hop QA (HotpotQA, MuSiQue, 2WikiMQA) with distractors and hides the true question behind a UUID chain. (3) **LongRLVR** (Chen et al., 2026): 18,870 QA examples from book, arXiv and code documents with 8K to 64K tokens, trained with an extra F_β-based context grounding reward over document chunks. A further comparison is shown in Table 6.

Benchmarks. We evaluate on five long-context benchmarks: (1) **AA-LCR** (Team, 2025a): 100 expert-crafted questions over real-world documents averaging 100K tokens. (2) **MRCR** (OpenAI, 2025): a multi-round coreference benchmark that asks the model to reproduce a specific response from a long dialog with repeated requests on overlapping topics. We evaluate with 2, 4, and 8 needles and report the average score. (3) **Frames** (Krishna et al., 2025): a multi-hop factual reasoning benchmark over multiple Wikipedia articles with numerical, temporal, or tabular reasoning. (4) **LongBench v2** (Bai et al., 2025): 503 multiple-choice questions over real documents from 8K to 2M words, covering QA, in-context learning, code and so on. (5) **LongReason** (Ling et al., 2025): a synthetic benchmark that turns short reasoning problems into long-context versions by spreading key information across 8K to 128K tokens. We evaluate at 8K, 16K, 32K, 64K, and 128K and report the average score.

We run AA-LCR 4 times and LongBench v2 2 times, and report the average; the other three benchmarks are run once.

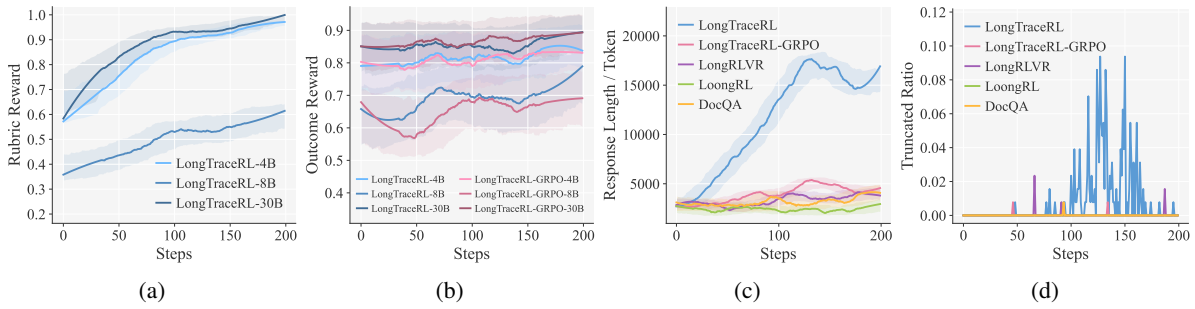
Training Details. We use the Slime framework (Zhu et al., 2025b) for RL training. The maximum context length is set to 160K tokens (128K prompt + 32K response). We use GRPO with group size $G = 8$, global batch size 128, and train for 200 iterations with a constant learning rate 2×10^{-6}. The rubric reward weight is $\alpha = 0.3$ by default, with group-level normalization and positive-only reward strategy. Rollout uses temperature 1.0, while all evaluations use temperature 0.6 and maximum generation length 32K. We save a checkpoint every 20 training steps and report results from the best-performing checkpoint. All experiments are conducted on 32 $\times$ H800 GPUs.

4.2 Main Results

Table 1 reports the per-benchmark scores of LongTraceRL against other baselines across three scales. LongTraceRL consistently achieves the best average score on every backbone: on Qwen3-4B-Thinking-2507 it reaches an average of **59.0**, improving the base model by **+5.7** points and surpassing the strongest baseline LongRLVR by **+2.5** points. The gain is most pronounced on the challenging AA-LCR (33.2 $\rightarrow$ 41.8, +8.6). The

Method	AA-LCR	MRCR	FRAMES	LongBench V2	LongReason	Avg
DeepSeek-R1-0528-Qwen3-8B						
Base	13.8	**27.0**	73.2	25.4	74.1	42.7
DocQA	9.5	20.2	73.4	25.9	73.7	40.6
LoongRL	10.2	19.0	72.6	25.5	73.3	40.1
LongRLVR	12.2	21.9	70.3	26.8	73.2	40.9
LongTraceRL-GRPO	**15.0**	25.2	73.1	25.6	**75.4**	42.9
LongTraceRL	**15.0**	**27.0**	**74.3**	**27.5**	75.2	**43.8**
Qwen3-4B-Thinking-2507						
Base	33.2	36.2	76.7	41.7	78.5	53.3
DocQA	28.8	41.9	78.3	**44.6**	79.9	54.7
LoongRL	32.0	38.2	75.8	41.8	78.7	53.3
LongRLVR	37.5	41.8	78.5	43.8	80.7	56.5
LongTraceRL-GRPO	34.0	38.9	76.1	40.7	78.7	53.7
LongTraceRL	**41.8**	**45.8**	**79.5**	44.1	**83.8**	**59.0**
Qwen3-30B-A3B-Thinking-2507						
Base	47.0	40.8	80.7	49.9	84.2	60.5
DocQA	50.2	44.8	**83.5**	51.6	86.4	63.3
LoongRL	**53.5**	43.8	81.9	**52.3**	84.9	63.3
LongRLVR	48.5	43.3	81.7	50.3	84.2	61.6
LongTraceRL-GRPO	48.2	44.7	79.6	51.8	**86.9**	62.3
LongTraceRL	**53.5**	**46.5**	81.9	51.3	85.4	**63.7**

Table 1: Main results on long-context reasoning benchmarks.



(a) (b) (c) (d)

Figure 3: From left to right: rubric and outcome reward dynamics at different scales, rollout response length and truncation rate dynamics across methods.

same trend holds for DeepSeek-R1-0528-Qwen3-8B (42.7 → 43.8) and Qwen3-30B-A3B-Thinking-2507 (60.5 → 63.7, +3.2), showing that the gains are robust to the model family and scale. In contrast, DocQA, LoongRL and LongRLVR even degrade performance on the 8B backbone (42.7 → 40.6 / 40.1 / 40.9). Besides, ablating the rubric reward (LongTraceRL-GRPO) on the 4B backbone drops the average score from 59.0 to 53.7, nearly erasing the gain despite training on the same dataset, identifying the rubric reward as the dominant driver of the improvement.

Figure 3a shows that the rubric reward grows steadily during LongTraceRL training on all three scales, indicating that the model progressively learns to ground its reasoning in the gold entities. Figure 3b further shows that the outcome reward curve of LongTraceRL also rises and dominates that of LongTraceRL-GRPO, confirming that introducing the rubric reward helps the model reach the correct final answer. Figure 3c compares rollout lengths across baselines, showing that the rubric reward clearly encourages longer and more deliberate reasoning. As shown in Figure 3d, around step 120, many rollouts of LongTraceRL reach the 32K budget and thus fail to emit the final answer, which suppresses the outcome reward. The positive-only strategy then guides the policy back to shorter responses before the length climbs again. This self-regulating behavior shows that combin-

Method	AA-LCR	MRCR	FRAMES	LongBench V2	LongReason	Avg
Qwen3-4B-Thinking-2507	33.2	36.2	76.7	41.7	78.5	53.3
+ LongTraceRL($\alpha = 0.1$)	39.2	**46.1**	79.0	**44.2**	82.8	58.3
+ LongTraceRL($\alpha = 0.3$)	**41.8**	45.8	**79.5**	44.1	**83.8**	**59.0**
+ LongTraceRL($\alpha = 0.5$)	39.0	43.7	77.5	43.5	81.7	57.1

Table 2: Performance of LongTraceRL with different rubric reward weight α.

Method	AA-LCR	MRCR	FRAMES	LongBench V2	LongReason	Avg
Qwen3-4B-Thinking-2507	33.2	36.2	76.7	41.7	78.5	53.3
+ LongTraceRL(`random`)	34.2	40.1	78.0	43.8	82.4	55.7
+ LongTraceRL(`search`)	34.2	43.2	79.0	43.9	83.3	56.7
+ LongTraceRL(`traj-random`)	35.5	44.2	**79.9**	**44.6**	82.8	57.4
+ LongTraceRL(`traj-tiered`)	**41.8**	**45.8**	79.5	44.1	**83.8**	**59.0**

Table 3: Performance of LongTraceRL across distractor strategies. Confusability increases from top to bottom.

ing the positive-only strategy with a finite response budget effectively prevents rubric reward hacking.

4.3 Ablation Studies

4.3.1 Rubric Ratio α

The hyperparameter α in Eq. 3 controls the relative weight of the rubric reward against the outcome reward, and thus determines how strongly the model is pushed to ground its reasoning in the gold entities. To study its effect, we sweep $\alpha \in \{0.1, 0.3, 0.5\}$ while keeping all other settings identical to our main experiment. As shown in Table 2, $\alpha = 0.3$ yields the best average score (59.0) and consistently achieves the best or second-best performance across all five benchmarks. Decreasing α to 0.1 weakens the process-level signal and degrades performance, especially on the more reasoning-intensive AA-LCR benchmark (41.8 $\rightarrow$ 39.2), while increasing α to 0.5 hurts performance across the board (average drops to 57.1), suggesting that an overly strong rubric weight begins to dilute the outcome objective and bias the model toward entity-mention shortcuts.

4.3.2 Source of Distractors

To verify the effectiveness of **`traj-tiered`** distractor strategy, we compare it against three alternative strategies while keeping the gold passages and questions identical: (1) **`random`**: distractors are randomly sampled from a global pool of documents retrieved across the dataset and are therefore mostly off-topic for the current question. (2) **`search`**: we issue the question as a single query to a search engine and leverage the top-100 results as distractors, approximating naive one-shot retrieval without multi-round querying, document opening or filtering. (3) **`traj-random`**: we pool Tier-1 and Tier-2 distractors of each question together and randomly sample documents from this pool, with no preference for their confusability.

This yields three new training sets that differ only in their distractor documents. We train Qwen3-4B-Thinking-2507 on each set under the same setting as the main experiment. As shown in Table 3, the `traj-tiered` strategy achieves the best average score (59.0), with the advantage particularly pronounced on AA-LCR (41.8 vs. at most 35.5 for the alternatives). To better explain these gains, we quantify distractor difficulty as the fraction of distractor documents that share at least one rubric entity with the reasoning chain and are thus harder to filter out. As Table 4 shows, the rank of distractor difficulty closely matches the rank of downstream scores in Table 3: `random` distractors almost never share rubric entities with the question (Macro Avg 1.35%), so they are easy to filter out and provide a weak training signal, leading to the lowest score (55.7). `search` strategy is slightly harder (Macro Avg 15.00%) and improves the score to 56.7, but it still cannot guarantee that the distractors lie along the reasoning path. Trajectory-based distractors are far more challenging: `traj-random` reaches 42.16%, and `traj-tiered` further pushes the ratio to 50.03%, with Tier-1 documents alone reaching 63.23%. This higher density of hard distractors aligns with the best downstream performance (59.0). These results confirm that the distractor is a key driver of long-context RL data quality, and that our tiered, trajectory-derived design is more effective than single-search-based or random-based alternatives.

Distractor Strategy	#Distr.	#w/ Rub.	Ent-Recall (%)	Micro Avg (%)	Macro Avg (%)
traj-tiered	62137	29050	**10.34**	**46.75**	**50.03**
Tier-1	18206	10629	14.65	58.38	63.23
Tier-2	43931	18421	8.56	41.93	41.75
traj-random	64066	26528	8.59	41.41	42.16
Tier-1	2677	1710	16.71	63.88	63.79
Tier-2	61389	24818	8.24	40.43	41.24
search	31412	4372	2.47	13.92	15.00
random	45392	565	0.16	1.24	1.35

Table 4: Statistics on how much distractors overlap with rubric entities. Higher ratios indicate harder distractors. **#Distr.**: number of distractor documents. **#w/ Rub.**: number of distractor documents containing $\geq$1 rubric entity. **Ent-Recall**: average fraction of rubric entities appearing in a distractor. **Micro/Macro Avg**: ratio of $\frac{\text{\#w/ Rub.}}{\text{\#Distr.}}$, aggregated globally (micro) or per sample then averaged (macro).

Method	AA-LCR	MRCR	FRAMES	LongBench V2	LongReason	Avg
Qwen3-4B-Thinking-2507	33.2	36.2	76.7	41.7	78.5	53.3
+ LongTraceRL(positive&negative)	37.0	40.5	79.5	**45.5**	83.1	57.1
+ LongTraceRL(positive-only)	**41.8**	**45.8**	**79.5**	44.1	**83.8**	**59.0**

Table 5: Performance of LongTraceRL with different reward strategies.

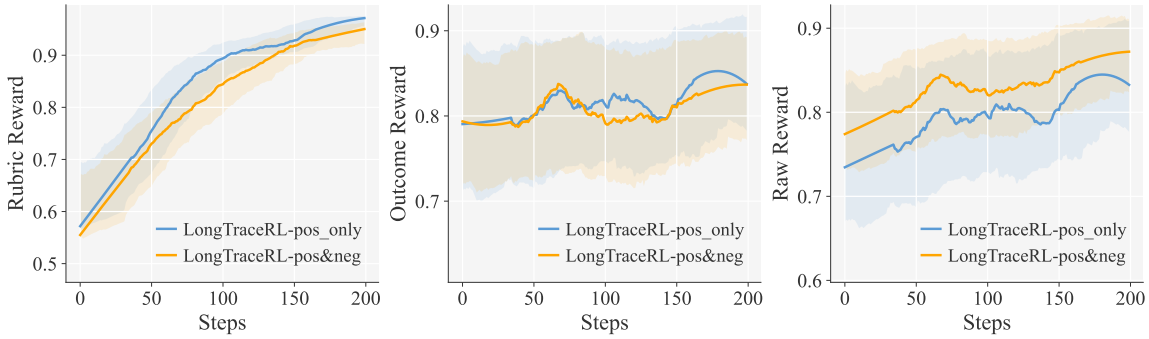


Figure 4: Rubric, outcome and combined raw reward dynamics for the two reward strategies.

4.3.3 Positive-Only

To verify the effectiveness of the **positive-only** strategy in preventing reward hacking, we compare it against a **positive&negative** variant in which the rubric reward is granted to every rollout regardless of answer correctness, i.e., $r = (1-\alpha)\cdot r_{\text{oc}} + \alpha \cdot r_{\text{rb}}$ for all responses. As shown in Table 5, removing the positive-only constraint causes a clear performance drop, with the average score falling from 59.0 to 57.1. The degradation is most pronounced on the reasoning-intensive benchmarks where the rubric signal is meant to help: AA-LCR drops by 4.8 points (41.8 $\rightarrow$ 37.0) and MRCR by 5.3 points (45.8 $\rightarrow$ 40.5). The training dynamics in Figure 4 further reveal the underlying mechanism. Throughout training, the positive&negative variant exhibits *lower* rubric and outcome rewards, yet its combined reward ends up *higher*, because every rollout, including the incorrect ones, can gain a non-trivial rubric term to the aggregate. This misleading objective dilutes the gradient toward genuinely solving the question and biases the policy toward enumerating gold-like entities from the context.

5 Conclusion

We present LongTraceRL, a framework that advances long-context RL by constructing challenging training data from trajectory-based distractors and introducing an entity-level rubric reward for fine-grained process supervision. Experiments on five long-context benchmarks across three model families and scales demonstrate consistent improvements over existing long-context RL methods. Further analysis confirms the effectiveness of each design choice. We hope that our approach offers a practical and generalizable recipe for further research on long-context RL of LLMs.

6 Limitations

Our work has several limitations. First, the data construction pipeline relies entirely on the KILT Wikipedia snapshot as its knowledge source, meaning all generated questions are grounded in encyclopedic knowledge. While our experiments show that training on such data transfers well to various downstream benchmarks covering financial, legal, and code documents, the single-source nature of the knowledge graph may limit the diversity of reasoning patterns in the training data. Second, the search agent trajectories used for distractor construction depend on the capabilities of the particular agent deployed. A stronger or weaker agent would produce different trajectory distributions, potentially affecting the quality and difficulty of the resulting distractors. Investigating how agent capability influences data quality is an interesting direction for future work.

7 Ethical Considerations

All models and datasets used in this work are publicly available under permissible licenses. Our method does not involve human subjects, private data, or content that raises dual-use concerns.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. 2024. Longalign: A recipe for long context alignment of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, Findings of ACL, pages 1376–1395. Association for Computational Linguistics.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2025. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 3639–3664. Association for Computational Linguistics.

Guanzheng Chen, Michael Qizhe Shieh, and Lidong Bing. 2026. Longrlvr: Long-context reinforcement learning requires verifiable context rewards. *CoRR*, abs/2603.02146.

Zhi Chen, Qiguang Chen, Libo Qin, Qipeng Guo, Haijun Lv, Yicheng Zou, Hang Yan, Kai Chen, and Dahua Lin. 2025. What are the essential factors in crafting effective long context multi-hop instruction datasets? insights and best practices. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 27129–27151. Association for Computational Linguistics.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Chaochen Gao, Xing Wu, Zijia Lin, Debing Zhang, and Songlin Hu. 2025. Nextlong: Toward effective long-context training without long documents. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.

Xin Guan, Zijian Li, Shen Huang, Pengjun Xie, Jingren Zhou, and Jiuxin Cao. 2026. Evidence-augmented policy optimization with reward co-evolution for long-context reasoning. *CoRR*, abs/2601.10306.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2025. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 4745–4759. Association for Computational Linguistics.

Siheng Li, Cheng Yang, Zesen Cheng, Lemao Liu, Mo Yu, Yujiu Yang, and Wai Lam. 2024. Large language models can self-improve in long-context reasoning. *CoRR*, abs/2411.08147.

Zhan Ling, Kang Liu, Kai Yan, Yifan Yang, Weijian Lin, Ting-Han Fan, Lingfeng Shen, Zhengyin Du, and Jiecao Chen. 2025. Longreason: A synthetic long-context reasoning benchmark via context expansion. *CoRR*, abs/2501.15089.

Rui Lu, Zhenyu Hou, Zihan Wang, Hanchen Zhang, Xiao Liu, Yujiang Li, Shi Feng, Jie Tang, and Yuxiao Dong. 2025. Deepdive: Advancing deep search agents with knowledge graphs and multi-turn RL. *CoRR*, abs/2509.10446.

OpenAI. 2025. OpenAI MRCR: Long context multiple needle in a haystack benchmark. Hugging Face dataset, inspired by the MRCR eval in Vodrahalli et al. (2024).

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies, pages 2523–2544, Online. Association for Computational Linguistics.

Bowen Ping, Zijun Chen, Yiyao Yu, Tingfeng Hui, Junchi Yan, and Baobao Chang. 2026. Longr: Unleashing long-context reasoning via reinforcement learning with dense utility rewards. *CoRR*, abs/2602.05758.

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.

Weizhou Shen, Ziyi Yang, Chenliang Li, Zhiyuan Lu, Miao Peng, Huashan Sun, Yingcheng Shi, Shengyi Liao, Shaopeng Lai, Bo Zhang, Dayiheng Liu, Fei Huang, Jingren Zhou, and Ming Yan. 2025. Qwenlong-l1.5: Post-training recipe for long-context reasoning and memory management. *CoRR*, abs/2512.12967.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Shreyas Singh, Kunal Singh, and Pradeep Moturi. 2025. Fathom-deepresearch: Unlocking long horizon information retrieval and synthesis for slms. *CoRR*, abs/2509.24107.

Artificial Analysis Team. 2025a. Artificial analysis long context reasoning benchmark(lcr).

Qwen Team. 2025b. Qwen3 technical report. *CoRR*, abs/2505.09388.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. 9835 musique: Multi-hop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554.

Kiran Vodrahalli, Santiago Ontanon, Nilesh Tripuraneni, Kelvin Xu, Sanil Jain, Rakesh Shivanna, Jeffrey Hui, Nishanth Dikkala, Mehran Kazemi, Bahare Fatemi, Rohan Anil, Ethan Dyer, Siamak Shakeri, Roopali Vij, Harsh Mehta, Vinay V. Ramasesh, Quoc Le, Ed H. Chi, Yifeng Lu, and 5 others. 2024. Michelangelo: Long context evaluations beyond haystacks via latent structure queries. *CoRR*, abs/2409.12640.

Fanqi Wan, Weizhou Shen, Shengyi Liao, Yingcheng Shi, Chenliang Li, Ziyi Yang, Ji Zhang, Fei Huang, Jingren Zhou, and Ming Yan. 2025. Qwenlong-l1: Towards long-context large reasoning models with reinforcement learning. *CoRR*, abs/2505.17667.

Siyuan Wang, Gaokai Zhang, Li Lyna Zhang, Ning Shang, Fan Yang, Dongyao Chen, and Mao Yang. 2025. Loongrl: Reinforcement learning for advanced reasoning over long contexts. *CoRR*, abs/2510.19363.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. Swe-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

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, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. 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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jiajie Zhang, Xin Lv, Ling Feng, Lei Hou, and Juanzi Li. 2026. Chaining the evidence: Robust reinforcement learning for deep search agents with citation-aware rubric rewards. *CoRR*, abs/2601.06021.

Dawei Zhu, Xiyu Wei, Guangxiang Zhao, Wenhao Wu, Haosheng Zou, Junfeng Ran, Xun Wang, Lin Sun, Xiangzheng Zhang, and Sujian Li. 2025a. Chain-of-thought matters: Improving long-context language models with reasoning path supervision. In *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 3197–3211. Association for Computational Linguistics.

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

A Dataset Comparison

Table 6 compares the training datasets of the long-context RL methods evaluated in our experiments. DocQA contains 200 MuSiQue examples (4-hop) and 200 MultiHopRAG examples (2–4 hop), and the remaining 1,191 are math or logic questions without a defined hop count. LoongRL reuses HotpotQA (2-hop), MuSiQue (2–4 hop), and 2WikiMQA (2-hop). Its KeyChain mechanism adds extra UUID-tracing steps on top. LongRLVR generates questions from single documents via LLM, which require cross-chunk synthesis but do not follow a defined multi-hop chain. In contrast, LONGTRACERL constructs questions via knowledge graph random walks with $k = 8$ hops, producing substantially deeper reasoning chains.

B Case Studies

Case from rollout data. Figure 5 shows a training rollout on a multi-hop question synthesized by our pipeline, whose reasoning chain spans seven gold entities (Arab-Berbers, Banu Hilal, Zenata, Marinid dynasty, Emirate of Granada, Granada War, Muhammad XII) before arriving at the final answer "Genil". A natural concern with our entity-recall rubric is that the model might learn to inflate its score by enumerating every entity it sees in the context. The rollout here illustrates the opposite behavior: LONGTRACERL-4B visits each gold entity in the correct order and commits to the final answer. Each cited entity serves a necessary role in the reasoning chain, and entities outside the gold path are never introduced merely to inflate the rubric score. This is exactly the behavior that the positive-only rubric reward is designed to elicit.

Case 1: Resolving Conflicting Cues in the Question. Figures 6 and 7 present a representative AA-LCR example that highlights the reasoning depth our reward formulation aims to promote. The question intentionally mixes two conflicting cues: its surface label says "medium-sized", but the supplied headcount of 450 employees actually falls into the ABS *large business* category (> 199 employees). The source document further reports two relevant ratios: a general 18.0% over *all* businesses, and a rate of 37.6% specifically for businesses with 200 to 999 workers. Therefore, to solve the question, one must first recognize this internal contradiction in the question and then retrieve the percentage that matches the actual headcount. As Figure 6 shows, LONGTRACERL-GRPO-4B takes a shortcut, selecting the first reasonable figure (18.0%) and missing the answer. In contrast, LONGTRACERL-4B trained with rubric reward (Figure 7) explicitly flags the inconsistency between "medium-sized" and 450 employees, reclassifies the firms by their actual headcount, and applies the 37.6% rate to get the correct answer. This example illustrates precisely how LONGTRACERL's design promotes deeper reading and reasoning.

Case 2: Disambiguating Pronoun Reference Across Clauses. Figures 8 and 9 illustrate a different type of reasoning challenge. In this question, two clauses share an ambiguous "this company": the first clause refers to a company that issued a press release on July 27, 2023 (Digital Realty), while the second clause introduces "one customer with an explicit mention of which stock exchange it trades on" and then asks about *that customer*'s exchange. Solving the question therefore requires (i) identifying the press-release issuer, (ii) realizing that the final "this company" refers back to the *customer* rather than the issuer, and (iii) finding the unique customer in the documents whose exchange is explicitly annotated (Equinix, Nasdaq: EQIX). As Figure 8 shows, LONGTRACERL-GRPO-4B merges the two clauses into a single entity, incorrectly treating Digital Realty as the answer. It actually retrieves the Equinix/Nasdaq passage but dismisses it because of a misapplied date constraint. In contrast, LONGTRACERL-4B (Figure 9) re-parses the question structure, explicitly pivots from issuer to customer, scans the documents for the unique customer with an exchange annotation, and returns the correct answer "Nasdaq".

Case 3: Understanding a Subtle Qualifier. Figures 10 and 11 highlight a third trap: a subtle filter hidden in the phrase "*within their scope*". The question asks for documents whose outlook covers the year 2022, which is an inclusive criterion: a document is qualified as long as its forecast period covers 2022, even when "2022" does not appear in its title. Two documents meet this requirement: the Deloitte "2022 Engineering and Construction Industry Outlook" (an obvious title match) and the ACEC "2021–2025 Engineering Industry Forecast", which provides explicit 2022 projections (e.g., "$356 billion in 2022") as part of its five-year forecast. As Figure 10 shows, LONGTRACERL-GRPO-4B secretly rewrites the question, focuses solely on the title-level keyword match and dis-

Dataset	#Samples	Context Length	#Hops	Question Source
DocQA	1,591	2K–20K	2–4	Real documents (math, logic, multi-hop)
LoongRL	15,000	16K	2–4	HotpotQA, MuSiQue, 2WikiMQA
LongRLVR	18,870	8K–64K	N/A	LLM-generated from book/arXiv/code
LongTraceRL	2,815	128K	8	KG random walk on Wikipedia

Table 6: Comparison of training datasets used in long-context RL methods.

cards the ACEC document despite having retrieved it. In contrast, LongTraceRL-4B (Figure 11) iterates over each candidate document, recognizes that the ACEC forecast covers 2022, and correctly returns both documents together with their source organizations.

C Prompts

We show our used prompts in Figure 12, 13.

Case from rollout data: Fully Solved by LongTraceRL-4B

Question: Begin with a mixed North African population whose identity emerged through cultural blending after eastern tribal migrations. One such migrating confederation moved west and weakened local nomadic groups, including a Berber confederacy that later produced a ruling house. From that house arose a dynasty that ruled a western Maghrebi realm and repeatedly intervened across a narrow sea to support the last Muslim polity there. That polity became the final Islamic state in the peninsula and was eventually destroyed in a prolonged campaign by united Christian crowns. The war ended when its final ruler formally yielded authority. Contemporary accounts place the ceremonial encounter of surrender beside a flowing watercourse near the city. What is the name of that watercourse? <DOC_1><DOC_2><DOC_3>……<DOC_N>

Answer: Genil

LongTraceRL-4B:

<think>

I need to follow the chain step by step and match each clue to the documents:

1. "Mixed North African population, identity emerged through cultural blending after eastern tribal migrations."
   → Doc [4] Arab-Berbers: mixed Berber/Arab origin, formed via Arab conquest and intermarriage (incl. Banu Hilal migration).

…

So the watercourse is the Genil.

</think>

Let me work through the chain step by step, matching each clue with the documents.

1. Mixed North African population from eastern migrations → **Arab-Berbers** [Doc 4]:

> "Arab-Berbers are people of mixed Berber, Arab and other non-Berber origin... The Arab-Berber identity came into being as a direct result of the Arab conquest of North Africa, and the intermarriage... Banu Hilal and Sulaym Arab tribes... intermarried with the local rural mainly Berber populations."

2. Migrating confederation that weakened a Berber confederacy → **Banu Hilal** weakened the **Zenata** [Doc 3]:

> "The Banu Hilal... emigrated to North Africa in the 11th century... they quickly defeated the Zirids and deeply weakened the neighboring Hammadid dynasty and the Zenata."

The Zenata [Doc 1] later produced a ruling house — the Marinids [Doc 5]: "a Sunni Muslim dynasty of Zenata Berber descent."

3. Dynasty ruling a western Maghrebi realm, intervening across a narrow sea → **Marinid dynasty** [Doc 5]:

> "ruled Morocco from the 13th to the 15th century... supported the Kingdom of Granada in Al-Andalus... attempt to gain a direct foothold on the European side of the Strait of Gibraltar."

4. Final Islamic state in the peninsula, destroyed by united Christian crowns → **Emirate of Granada**, destroyed in the **Granada War** [Doc 10]:

> "between 1482 and 1491, during the reign of the Catholic Monarchs Isabella I of Castile and Ferdinand II of Aragon... ending all Islamic rule on the Iberian peninsula."

5. Final ruler formally yielding authority → **Muhammad XII (Boabdil)** [Doc 2]:

> "In January 1492 Muhammad XII of Granada, the last Nasrid ruler... formally relinquished his sovereignty and surrendered his territories to Castile."

6. Watercourse at the ceremonial surrender → From Doc [7], eyewitness letter to the bishop of León:

> "the royal procession moved from Santa Fe to a place a little more than a mile from Granada, where Ferdinand took up his position by the banks of the Genil."

Doc [6] confirms the **Genil** flows through Granada.

Final Answer: Genil (Genil River) ✔

Figure 5: A training rollout from LongTraceRL-4B on a synthesized multi-hop question.

Case 1 on AA-LCR: Failed by LongTraceRL-GRPO-4B

Conflicts!

Question: Consider a sample of 560 **medium-sized** Australian business each with around **450 employees**. How many of these businesses (to the nearest whole number) would we expect to have employees that are restrained by co-worker non-solicitation clauses? <DOC_1><DOC_2><DOC_3>.....<DOC_N>
Answer: 211

LongTraceRL-GRPO-4B:
To answer this question, I need to determine how many of the 560 medium-sized Australian businesses with around 450 employees would be expected to have employees restrained by co-worker non-solicitation clauses.

From Document 1 (the Issues Paper), I can find the relevant statistic: **"The ABS survey reported 18.0 per cent of businesses used a co-worker non-solicitation clause for some of their workers in 2023."**

This is the key figure I need for my calculation. The document also notes that "The ABS survey also reported that 67.2 per cent of businesses which used co-worker non-solicitation clauses, used them for over three-quarters of their workers" - but this is about the percentage of workers affected within businesses that use these clauses, not the percentage of businesses that use them.

For the calculation:
- Total businesses: 560
- Percentage of businesses with co-worker non-solicitation clauses: 18.0%
- Expected number of businesses: 560 × 0.18 = 100.8

Rounded to the nearest whole number, this equals 101 businesses.

The question asks specifically about how many businesses would have employees restrained by co-worker non-solicitation clauses, not how many employees are affected. Since 18.0% of businesses use co-worker non-solicitation clauses (for some of their workers), the answer is 101 businesses.

Therefore, we would expect 101 out of the 560 medium-sized Australian businesses with around 450 employees each to have employees restrained by co-worker non-solicitation clauses.

Answer: 101 ✗

Figure 6: A failure case from AA-LCR where the LongTraceRL-GRPO-4B trained without rubric reward takes a shortcut without resolving the conflict in the question.

Case 1 on AA-LCR: Fully Solved by LongTraceRL-4B

Conflicts!

Question: Consider a sample of 560 **medium-sized** Australian business each with around **450 employees**. How many of these businesses (to the nearest whole number) would we expect to have employees that are restrained by co-worker non-solicitation clauses? <DOC_1><DOC_2><DOC_3>……<DOC_N>
Answer: 211

LongTraceRL-4B:
To answer how many of 560 Australian businesses (each with ~450 employees) would be expected to have employees restrained by co-worker non-solicitation clauses, I need to examine the input documents for relevant evidence.

Step 1: Clarify business size definitions
From Document 1 (Issues Paper):
> "The ABS defines business size based on employment size, grouped into micro/small business (0-19 employees), medium-sized business (20-199 employees) and large business (employment size greater than 199 employees)."
So under the ABS classification:
- Micro/small: 0–19 employees
- Medium-sized: 20–199 employees
- Large: >199 employees
The question calls the sample "medium-sized" but each business has ~450 employees, which actually falls in the **large** category (specifically the 200–999 worker bracket). **This appears to be a misnomer in the question — I'll use the bracket matching the employee count.**

Step 2: Find the prevalence of co-worker non-solicitation clauses
From Document 1:
> "The ABS survey reported 18.0 per cent of businesses used a co-worker non-solicitation clause for some of their workers in 2023, [37] the least used of the four restraint clauses reported in the survey."
> "Co-worker non-solicitation clauses were most often used by large businesses, with reported use of 37.7 per cent among businesses that have 1,000 or more workers and **37.6 per cent for businesses with between 200 and 999 workers**."
This is the critical figure — **450-employee businesses fall in the 200–999 bracket, so the applicable rate is 37.6%.**

Step 3: Confirm "restrained employees" interpretation
Also from Document 1:
> "The ABS survey also reported that 67.2 per cent of businesses which used co-worker non-solicitation clauses, used them for over three-quarters of their workers."
This confirms that businesses using the clauses do in fact have restrained employees.

Step 4: Cross-check against other documents
From Document 9 (COSBOA submission), the 18.0% overall figure is repeated, but this is the all-business-size average — not specific to 200–999 workers. **The 37.6% figure from Document 1 remains the correct one for this sample.**

Calculation:
$560 \times 0.376 = 210.56 \approx 211$

Answer: 211

Figure 7: A success case from AA-LCR where the LongTraceRL-4B trained with rubric reward identifies the conflict and applies the correct rate (37.6%) to reach the correct answer.

Case 2 on AA-LCR: Failed by LongTraceRL-GRPO-4B

Question: **One company** in the documents conducted a press release on July 27, 2023. There is only **one customer** with an explicit mention of which stock exchange it trades on within the document set. According to the documents, what stock exchange does does **this company** trade on? <DOC_1><DOC_2><DOC_3>……<DOC_N>

Answer: Nasdaq

refers to the "one customer" instead of "one company"

LongTraceRL-GRPO-4B:

The question specifies there is "only one customer with an explicit mention of which stock exchange it trades on" within the document set. Looking through all the documents, **Digital Realty is the only company that has an explicit mention of which stock exchange it trades on - it trades on the NYSE (New York Stock Exchange) with the ticker symbol DLR.**

While Document 3 mentions Equinix (Nasdaq: EQIX), that press release was dated August 2, 2023, not July 27, 2023, so it doesn't meet the specific date requirement in the question.

Therefore, according to the documents, the company that conducted a press release on July 27, 2023 and has an explicit mention of which stock exchange it trades on trades on the NYSE (New York Stock Exchange).

Answer: NYSE ✗

Figure 8: A failure case from AA-LCR where the LongTraceRL-GRPO-4B trained without rubric reward incorrectly merges the two clauses into a single entity and answers with the press-release issuer's exchange.

Case 2 on AA-LCR: Fully Solved by LongTraceRL-4B

Question: **One company** in the documents conducted a press release on July 27, 2023. There is only **one customer** with an explicit mention of which stock exchange it trades on within the document set. According to the documents, what stock exchange does does **this company** trade on? <DOC_1><DOC_2><DOC_3>……<DOC_N>
Answer: Nasdaq

refers to the "one customer" instead of "one company"

LongTraceRL-4B:
To answer the question, I need to (1) identify the company that issued a press release on July 27, 2023, (2) find the only customer in the document set with an explicit stock-exchange mention.

Step 1: Company with a press release on July 27, 2023
- **Document 1**: "Digital Realty Reports Second Quarter 2023 Results" — "Austin, TX — July 27, 2023 — Digital Realty (NYSE: DLR)…"
- **Document 2**: Same Digital Realty press release, dated July 27, 2023.
- **Document 3**: Equinix press release dated August 2, 2023 — not a match.

→ The company is **Digital Realty Trust, Inc. (NYSE: DLR)**.

Step 2: Customer with an explicit stock-exchange mention
Document 1 contains Digital Realty's "Top 20 Customers by Annualized Rent" table, including names such as Oracle Corporation, IBM, Equinix, LinkedIn, Meta Platforms, JPMorgan Chase, Comcast, AT&T, Zayo, Lumen, Rackspace, Cyxtera, etc. The table itself does not mention any stock exchanges.
Searching for explicit exchange mentions tied to these customers across the documents:
- **Document 3 ("About Equinix")** states: ***"Equinix (Nasdaq: EQIX) is the world's digital infrastructure company."*** — explicit exchange mention.
- All other listed customers (Oracle, IBM, Meta, JPMorgan Chase, AT&T, Zayo, etc.) are referenced by name only; **no explicit stock exchange is stated** for them anywhere in the document set.

Equinix is the only customer with an explicit stock-exchange mention, and it appears in Digital Realty's customer table (rank 7).

Step 3: Confirm uniqueness
No other customer in the documents has an explicit exchange identifier — only Equinix's "Nasdaq: EQIX" qualifies.

Answer: **Nasdaq** ✔

Figure 9: A success case from AA-LCR where the LongTraceRL-4B trained with rubric reward keeps the two clauses separate and returns the correct answer.

Case 3 on AA-LCR: Failed by LongTraceRL-GRPO-4B

Question: I am researching construction trends. I am looking for documents that **provide outlooks for the year 2022 within their scope**. Which documents should I read? Please outline the full title of the document and which organisation it is from. `<DOC_1><DOC_2><DOC_3>.....<DOC_N>`
Answer: Deloitte: 2022 Engineering and Construction Industry Outlook, ACEC Research Institute: 2021-2025 Engineering Industry Forecast.

LongTraceRL-GRPO-4B:
Based on your request for documents that provide outlooks specifically for the year 2022 within their scope, I've identified the most relevant document from the input materials:

Document Title: "2022 engineering and construction industry outlook"
Organization: Deloitte

This document explicitly focuses on the outlook for the year 2022 and provides detailed analysis of the engineering and construction industry's expected performance, challenges, and opportunities for that specific year. The document includes sections specifically discussing the industry's position for 2022, market dynamics, profitability challenges, digital transformation trends, and M&A activity that would impact the industry in that year.

The document is clearly labeled with "2022" in its title and contains forward-looking statements specifically about what the industry can expect in 2022, making it the most direct match for your research needs.

While Document 5 ("2021-2025 Engineering Industry Forecast" by ACEC Research Institute) does include 2022 in its multi-year forecast, it's a broader 5-year forecast rather than a dedicated 2022 outlook. The Deloitte document is the most focused specifically on 2022 as the year of interest.

This document would be your best starting point for understanding the construction industry's outlook specifically for 2022.

Answer: Deloitte: 2022 Engineering and Construction Industry Outlook

Figure 10: A failure case from AA-LCR where the LongTraceRL-GRPO-4B trained without rubric reward silently rewrites the question and uses a tightened criterion to discard the ACEC document.

Case 3 on AA-LCR: Fully Solved by LongTraceRL-4B

Question: I am researching construction trends. I am looking for documents that **provide outlooks for the year 2022 within their scope**. Which documents should I read? Please outline the full title of the document and which organisation it is from. <DOC_1><DOC_2><DOC_3>……<DOC_N>

Answer: Deloitte: 2022 Engineering and Construction Industry Outlook, ACEC Research Institute: 2021-2025 Engineering Industry Forecast.

LongTraceRL-4B:

I reviewed all 11 documents to identify those that provide outlooks specifically for 2022 within their scope.

- Doc 1 — KPMG "Construction in 2030" (Feb 2023): a 2030 projection. **Not 2022.**
- Doc 2 — "No turning back" (2021 Global Construction Survey, Aug 2021): 2021 survey, not a 2022-specific outlook. **Not 2022.**
- Doc 3 — Deloitte "2024 engineering and construction industry outlook" (Nov 2023): explicitly a 2024 outlook. **Not 2022.**
- Doc 4 — "2020 engineering and construction": outlook for 2020. **Not 2022.**
- Doc 5 — ACEC Research Institute, **"2021-2025 Engineering Industry Forecast"**: 5-year forecast covering 2022 explicitly, e.g. *"The Engineering and Design Services sector… will begin to recover over the period, growing 3% to $356 billion in 2022."* → **2022 outlook** ✓
- Doc 6 — "2023 engineering and construction": outlook for 2023, with 2022 only as backward context. **Not 2022.**
- Doc 7 — "The next normal in construction" (June 2020): a 2020 report with forward-looking statements; not 2022-specific. **Not 2022.**
- Doc 8 — McKinsey "From start-up to scale-up" (2023): construction tech trends, not a 2022 outlook. **Not 2022.**
- Doc 9 — Deloitte Research Center for Energy & Industrials, **"2022 engineering and construction industry outlook"**: explicitly titled as 2022 outlook ("Preparing for another strong year"; *"In 2022, as we move into the second year of recovery…"*). → **2022 outlook** ✓
- Doc 10 — Deloitte "2021 engineering and construction industry outlook" (Nov 2021): outlook for 2021. **Not 2022.**
- Doc 11 — KPMG "Familiar challenges—new approaches" (2023 Global Construction Survey, June 2023): 2023 survey. **Not 2022.**

Final Answer

Two documents provide outlooks for 2022 within their scope:

1. **"2021-2025 Engineering Industry Forecast"** — *ACEC Research Institute* (Doc 5): 5-year forecast explicitly covering 2022.
2. **"2022 engineering and construction industry outlook"** — *Deloitte Research Center for Energy & Industrials* (Doc 9): explicitly a 2022 outlook.

Figure 11: A success case from AA-LCR where the LongTraceRL-4B trained with rubric reward checks each candidate document and finds both qualifying outlooks with correct organization attributions.

Prompt for Multi-hop QA Generation

Task Description

You are provided with:

- A path: an ordered sequence of entities (nodes) obtained from a random walk on Wikipedia, where each node is connected by a specific relationship.
- Descriptions of each node: detailing key attributes such as names, places, events, or times.

Your task:

- Find the connections among nodes from the path, and construct a multi-hop question as hard as possible that requires reasoning along the path step by step, using each node and relationship in order to arrive at the solution.
- Select one attribute (could be a name, place, event, or time) from the description of the last node in the path as the **final unique answer** to the question.

- **CRITICAL**: When constructing clues, focus on the **relationship and connection** to the previous step, rather than providing intrinsic attributes of the node itself. Do NOT providing unique or distinguishing features that would allow direct identification of a node without reasoning through the previous steps. The clues MUST guide the reasoning along the path step by step, not enable shortcuts.
- Any obvious, distinct, or detailed clues MUST NOT show up in the question, such as names, places, times, or events. Instead, paraphrase or generalize all clues and relationships, so that the answer cannot be obtained by simple keyword matching.
- Every node's information should be mentioned in the question.
- The question should include up to `{num_entities * 2}` unique clues.
- The question should be concise and coherent, with a total character count **not exceeding** `{num_entities * 50}`.
- The answer MUST be unique, and exactly match the selected attribute from the last node, ensuring there is no other answer that satisfies the question.

Output Format

```
<question>(A multi-step question, arranged in the order of the path, in English)</question>
<answer>(The unique answer corresponding to the selected attribute from the last node)</answer>
<entities>(A python list of all entities needed along the reasoning path)</entities>
```

Input

- Path: `{path}`
- Node Descriptions: `{intro}`

Figure 12: Prompt for multi-hop QA generation in LongTraceRL.

Prompt for Outcome Reward Judgement

Judge whether the following **[response]** to **[question]** is correct or not based on the precise and unambiguous **[correct_answer]** below.

[question]: `{question}`

[correct_answer]: `{answer}`

[response]: `{response}`

Your judgement must be in the format and criteria specified below:

- **extracted_final_answer**: The final exact answer extracted from the **[response]**. Put the extracted answer as **"None"** if there is no exact, final answer to extract from the response.
- **reasoning**: Explain why the extracted_final_answer is correct or incorrect based on **[correct_answer]**, focusing only on if there are meaningful differences between **[correct_answer]** and the **extracted_final_answer**. Do not comment on any background to the problem, do not attempt to solve the problem, do not argue for any answer different than **[correct_answer]**, focus only on whether the answers match
- **correct**: Answer **"yes"** if **extracted_final_answer** matches the **[correct_answer]** given above, or is within a small margin of error for numerical problems. Answer **"no"** otherwise, i.e. if there if there is any inconsistency, ambiguity, non-equivalency, or if the extracted answer is incorrect.

Figure 13: Prompt for outcome reward judgement in LongTraceRL.