Learning to Localize and Repair Deviations in Long-Horizon Reasoning

Anonymous ACL submission

Abstract

Large language model (LLM) agents are increasingly used to solve long-horizon tasks through reasoning, tool use, and environment interaction. Yet each additional step introduces fresh opportunities for local errors, which compound across subsequent reasoning and derail the trajectory. Existing feedback mechanisms predominantly provide scalar supervision over intermediate behaviors or revision-oriented feedback at the level of complete trajectories, limiting their capacity for localized correction and repair. We therefore propose **RLOR**, an observation-guided trajectory repair framework that treats long-horizon reasoning as an intervenable process. RLOR introduces pivots to localize deviations, injects observation-based pivot reasons as process-level feedback, and repairs only the subsequent continuation while preserving the valid prefix. The framework combines curriculum Supervised Fine-Tuning (SFT) with pivot-aware Reinforcement Learning (RL) to learn direct solving, pivot triggering when needed, and observation-conditioned suffix repair. Experiments on long-horizon reasoning benchmarks demonstrate that RLOR improves both accuracy and token efficiency, with ablations confirming the contributions of pivot localization, observation content, curriculum SFT, and pivot-aware RL with credit assignment.

1 Introduction

In recent years, large language models (LLMs) have evolved from static text generators into autonomous agents capable of tool use, environment interaction, and multi-step decision making (Nakano et al., 2021; Yao et al., 2022; Schick et al., 2023; Trivedi et al., 2023; Qin et al., 2024; Liu et al., 2024b). This paradigm shift has underscored the critical importance of long-horizon reasoning. In complex domains such as multi-hop question answering and tool-augmented problem solving, models must maintain task goals, integrate external evidence, and adapt their action strategies across multiple rounds of tool-use and reasoning. However, the central challenge in such settings is the compounding nature of intermediate errors. Once an agent forms an incorrect hypothesis or overlooks a key constraint, subsequent reasoning may continue to unfold around this mistaken direction. Consequently, simply extending the context window or allocating additional inference steps does not necessarily improve performance; it may instead prolong exploration within an invalid search space (Liu et al., 2024a; Huang et al., 2024). Effective long-horizon reasoning, therefore, requires mechanisms that can maintain trajectory reliability and recover from intermediate deviations.

Prior work has explored finer-grained supervision and feedback to improve intermediate reasoning. Process-supervision and verifier-based methods assign granular labels or reward scores to intermediate steps, providing denser training signals than final-answer supervision (Lightman et al., 2024; Wang et al., 2024; Shao et al., 2025). While beneficial for step-wise verification and credit assignment, these scalar signals depend on the quality and alignment of the verifier and contain limited explicit information for correction. Natural-language feedback and self-refinement methods provide richer corrective signals through critiques, reflections, or tool-grounded validation (Scheurer et al., 2023; Madaan et al., 2023; Shinn et al., 2023; Gou et al., 2024). Yet because these critiques are typically produced after a complete output, they predominantly drive holistic trajectory revisions rather than targeted correction of the erroneous segment.

To this end, we propose RLOR, an observation-guided trajectory repair framework for long-horizon reasoning. Our key insight is that long-horizon reasoning should be treated as an intervenable process rather than a sequence optimized only by terminal feedback. When a trajectory deviates, RLOR

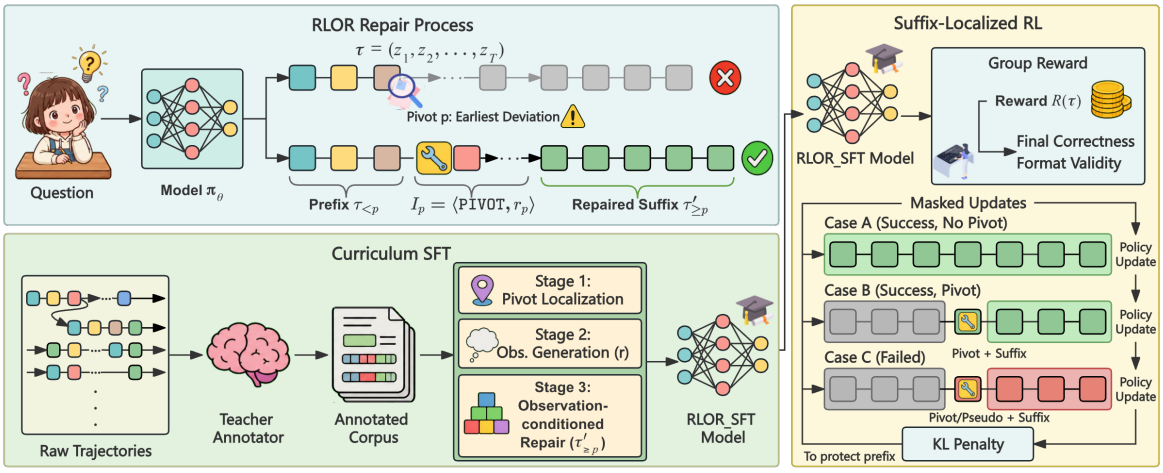

Figure 1: Overview of RLOR and its training pipeline. RLOR repairs trajectory deviations in long-horizon reasoning by localizing the earliest deviation as a pivot, injecting an observation-based pivot reason, preserving the valid prefix, and generating a repaired suffix. The model is trained through curriculum supervised fine-tuning for pivot localization, observation-based pivot-reason generation, and continuation repair, followed by suffix-localized reinforcement learning that encourages direct solving and pivot-triggered repair when needed.

avoids both restarting from scratch and continuing along an unreliable path by localizing the deviation and repairing the trajectory from that point onward. RLOR introduces a pivot to mark the earliest step where the trajectory departs from a valid reasoning path, preserves the valid prefix before the pivot, and injects an observation-based pivot reason that diagnoses error or reinstates neglected task constraints. The model then generates a repaired suffix conditioned on the preserved prefix and pivot reasons, coupling failure localization, process-level feedback conditioning, and localized continuation repair.

We evaluate RLOR on long-horizon reasoning benchmarks covering multi-step question answering and tool-augmented reasoning. Extensive experimental results demonstrate that RLOR improves both final accuracy and token efficiency over strong baselines. Further ablations analyze the roles of pivot localization, observation content, curriculum Supervised Fine-Tuning (SFT), and Reinforcement Learning (RL) with suffix-localized credit assignment. These results indicate that RLOR's gains do not merely come from longer reasoning or additional supervision, but from learning a structured repair behavior that preserves reliable computation while correcting unreliable trajectory segments.

Our contributions are summarized as follows:

- We formulate long-horizon reasoning optimization as observation-guided trajectory repair, introducing pivots and observation-based pivot reasons to preserve valid prefixes and repair deviated continuations.

- We propose RLOR, a training framework that combines curriculum SFT with pivot-aware RL, enabling models to solve directly, trigger pivots when needed, and repair suffixes conditioned on pivot reasons.

- We show that RLOR improves both accuracy and token efficiency on long-horizon reasoning benchmarks, with ablation studies confirming the contributions of pivot localization, observation content, curriculum SFT and RL with suffix-localized credit assignment.

2 RLOR: Reinforcement Learning for Observation-guided Reasoning Repair

We propose RLOR, an observation-guided training framework that improves long-horizon reasoning through localized trajectory repair. RLOR identifies the earliest deviation as a pivot, preserves the valid prefix, and injects a process observation as the pivot reason. The model then continues from pivot reason by generating a repaired suffix, reusing reliable prior reasoning while correcting the post-deviation continuation.

2.1 Formalization of Reasoning Trajectories and Pivots

Given a problem instance x, the model generates a trajectory $\tau = (z_1, z_2, \ldots, z_T)$, where each step z_t encapsulates internal reasoning, tool invocations, tool observations, or the final answer. In tool-augmented settings, the trajectory follows an interleaved reasoning–acting process, where subsequent decisions are conditioned on prior reasoning states and external environmental feedback.

We characterize the implicit reasoning state at step t as $S_t = (K_t, H_t, G_t)$, where K_t denotes verified facts or intermediate conclusions, H_t denotes candidate hypotheses or reasoning branches under evaluation, and G_t denotes the task objective and constraints. This decomposition distinguishes what has been established, what remains exploratory, and what conditions the final solution must satisfy.

RLOR introduces a pivot variable p to demarcate the earliest step at which the trajectory departs from a solvable reasoning state:

$$p = \min\{t \mid S_t \not\rightsquigarrow \mathcal{S}^\star(x)\},$$

where $\mathcal{S}^\star(x)$ is the set of implicit states from which the task remains solvable under valid reasoning policy. A pivot therefore marks the earliest boundary at which trajectory should transition from direct solving mode to repair mode. If no such step exists, the trajectory is treated as a no-pivot trajectory.

2.2 Observation-based Pivot Intervention

Given a pivot p, RLOR preserves the prefix

$$\tau_{<p} = (z_1, \ldots, z_{p-1})$$

and appends an intervention package $I_p = \langle \texttt{PIVOT}, r_p \rangle$, where r_p is an observation-based pivot reason. The repaired suffix is sampled as

$$\tau'_{\geq p} \sim \pi_\theta(\cdot \mid x, \tau_{<p}, I_p),$$

yielding the observation-augmented trajectory

$$\tilde{\tau} = (\tau_{<p}, I_p, \tau'_{\geq p}).$$

The pivot reason r_p serves as process-level feedback and takes one of two forms. An *evaluative observation* diagnoses the local failure, such as an unsupported assumption or erroneous reasoning branch. A *contextual observation* re-exposes task constraints or verified intermediate conclusions. Thus, the intervention changes the conditional distribution of future reasoning without rewriting the valid prefix.

2.3 Curriculum SFT for Pivot Repair

We first construct teacher-annotated curriculum data from raw reasoning trajectories. Given $\mathcal{T} = \{(x_i, \tau_i)\}_{i=1}^N$, we use GPT-5.2 (OpenAI, 2025d) to annotate whether each trajectory deviates from a valid reasoning path. If a deviation exists, it identifies the earliest pivot position $\hat{p}_i$, extracts the valid prefix $\hat{\tau}_{i,<\hat{p}_i}$, generates an observation-based pivot reason $\hat{r}_i$, and produces a repaired continuation $\hat{\tau}'_{i,\geq\hat{p}_i}$ conditioned on the prefix and pivot intervention. This yields the annotated corpus:

$$\mathcal{D}_{\text{ann}} = \left\{ \left(x_i, \hat{\tau}_{i,<\hat{p}_i}, \hat{p}_i, \hat{r}_i, \hat{\tau}'_{i,\geq\hat{p}_i} \right) \right\}_{i=1}^N,$$

From $\mathcal{D}_{\text{ann}}$, we construct curriculum SFT datasets that decompose pivot repair into three subtasks: pivot localization, pivot reason generation, and pivot-conditioned continuation. The first stage trains the model to identify where local intervention is needed. The second stage trains the model to generate a pivot marker together with an evaluative or contextual observation. The third stage trains the model to produce a repaired suffix conditioned on the original prefix and pivot intervention.

For each stage k, we optimize the standard autoregressive objective

$$\mathcal{L}_{\text{sft}}^{(k)}(\theta) = -\mathbb{E}_{(u,y)\sim\mathcal{D}^{(k)}} \sum_{j=1}^{|y|} \log \pi_\theta(y_j \mid u, y_{<j}),$$

where u and y denote the stage-specific input and target output. Each stage is initialized from the checkpoint of the preceding stage.

2.4 Reinforcement Learning with Suffix-Localized Credit Assignment

After curriculum SFT, RLOR further optimizes the model with reinforcement learning to balance direct solving and localized repair. Given a prompt x, we sample a group of G trajectories from the old policy: $\{\tau_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$. Each trajectory is evaluated by final correctness $a_i \in \{0, 1\}$ and format validity $f_i \in \{0, 1\}$. We use the scalar reward

$$R(\tau_i) = \begin{cases} 1, & a_i = 1 \text{ and } f_i = 1, \\ 0.1, & a_i = 0 \text{ and } f_i = 1, \\ 0, & f_i = 0, \end{cases}$$

and compute the group-normalized advantage

$$\hat{A}_i = \frac{R(\tau_i) - \mu_R}{\sigma_R + \epsilon},$$

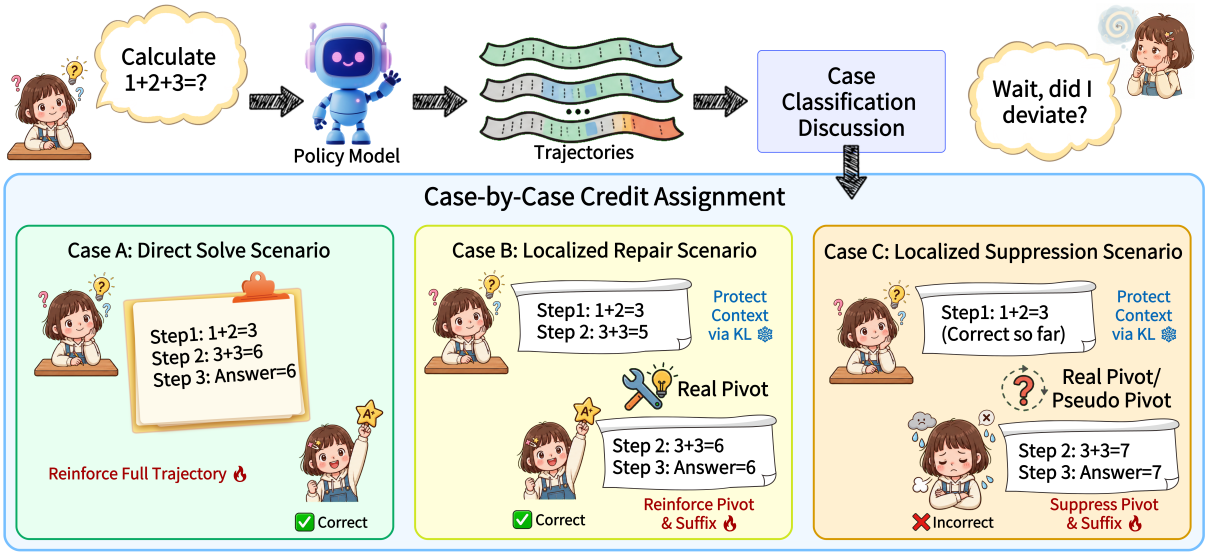

Figure 2: Case-by-case credit assignment in suffix-localized RL. RLOR classifies generated trajectories into direct-solve, localized-repair, and localized-suppression cases according to final correctness and pivot behavior. Correct no-pivot trajectories reinforce full direct solving; correct trajectories with pivot reinforce only the pivot and repaired suffix; incorrect trajectories suppress the pivot or pseudo-pivot and suffix while protecting the valid prefix.

where μ_R and σ_R are the mean and standard deviation of rewards within the group.

As is shown in Figure 2, RLOR applies different credit-assignment masks according to trajectory type. Let Ω_i denote the tokens receiving the main policy-gradient update, and Ψ_i denote tokens subject to weak KL regularization.

Direct-Solve (Correct, no pivot). For a correct no-pivot trajectory, RLOR reinforces the entire assistant trajectory:

$$\Omega_i = \text{Tokens}(\tau_i), \qquad \Psi_i = \emptyset.$$

Localized-Repair (correct, with pivot p_i). The main update is restricted to the pivot region and the repaired suffix:

$$\Omega_i = \text{Tokens}(\tau_{i,\geq p_i}), \qquad \Psi_i = \text{Tokens}(\tau_{i,<p_i}).$$

Localized-Suppression (incorrect). For an incorrect trajectory, RLOR treats it as requiring repair. If a pivot exists, it is used as the repair boundary; otherwise, a pseudo-pivot $\tilde{p}_i$ is introduced by selecting the step with the highest policy entropy as an uncertainty-based proxy for the deviation point, motivated by (Wang et al., 2026):

$$b_i = \begin{cases} p_i, & \text{if a pivot exists in } \tau_i, \\ \tilde{p}_i, & \text{otherwise.} \end{cases}$$

The update mask is then set as follows:

$$\Omega_i = \text{Tokens}(\tau_{i,\geq b_i}), \qquad \Psi_i = \text{Tokens}(\tau_{i,<b_i}).$$

We optimize Ω_i with a masked sequence-level GSPO (Zheng et al., 2025) objective. For each trajectory τ_i, we compute the group-normalized advantage $\hat{A}_i$ from the rewards of G sampled trajectories for the same input. For token y_j with autoregressive context c_j, the length-normalized masked sequence-level ratio is defined as

$$\rho_i(\theta) = \exp\left(\frac{1}{|\Omega_i|}\sum_{j\in\Omega_i}\log\frac{\pi_\theta(y_j \mid c_j)}{\pi_{\theta_{\text{old}}}(y_j \mid c_j)}\right).$$

Given $\rho_i(\theta)$ and $\hat{A}_i$, we define the clipped surrogate

$$g(\rho, A) = \min\left(\rho A,\ \text{clip}(\rho, 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}})A\right).$$

The main objective is

$$\mathcal{J}_{\text{main}}(\theta) = \mathbb{E}_x\left[\frac{1}{G}\sum_{i=1}^{G} g\big(\rho_i(\theta), \hat{A}_i\big)\right].$$

To protect valid prefixes from being perturbed by the current RL update, RLOR excludes prefix tokens Ψ_i from the main reward-driven objective and applies a weak KL penalty against the old policy:

$$\mathcal{L}_{\text{KL}} = \mathbb{E}_i\left[\frac{1}{|\Psi_i|}\sum_{j\in\Psi_i} D_{\text{KL}}\left(\pi_\theta(\cdot \mid c_j) \| \pi_{\theta_{\text{old}}}(\cdot \mid c_j)\right)\right].$$

The final objective is defined as:

$$\max_\theta \quad \mathcal{J}_{\text{RLOR}}(\theta) = \mathcal{J}_{\text{main}}(\theta) - \beta\mathcal{L}_{\text{KL}},$$

Table 1: Main results on five long-horizon agent benchmarks. RLOR is compared with same-backbone baselines built on Qwen3.5-35B-A3B and representative proprietary/open-source agent systems. Tool Use indicates whether the model invokes external tools, including Search and LinkSummary, during inference. The best result within each block is highlighted in bold.

Type	Model	Tool Use	Browse Comp	AMO_Bench	SealQA_0	HLE	GAIA
Direct Inference	GPT-4.1	✗	1.5	4.1	–	4.9	22.3
	o4-mini	✗	**6.1**	**40.2**	4.5	**18.9**	**33.3**
	DeepSeek-R1-0528	✗	2.0	34.3	**5.4**	17.7	16.5
Proprietary Agents	OpenAI DeepResearch	✓	51.5	–	–	26.6	67.4
	Kimi-Researcher	✓	–	–	36.0	26.9	–
	GPT-5	✓	54.9	52.4	**51.4**	35.2	**76.4**
	Gemini-3.0-Pro	✓	**59.2**	**63.1**	45.5	**46.9**	–
Open-Source Agents	Asearcher-Web	✓	5.2	–	–	12.5	52.8
	GLM-4.5 Heavy	✓	**54.0**	–	–	–	66.0
	Tongyi DeepResearch	✓	43.4	–	–	**32.9**	**70.9**
RLOR (Backbone: Qwen3.5-35B-A3B)	Mono	✓	32.7	32.0	27.1	40.1	69.2
	Mixture-SFT	✓	33.3	34.0	28.5	39.6	70.9
	Stage-SFT	✓	35.3	54.0	32.5	40.5	73.1
	GSPO	✓	43.6	60.0	40.3	46.8	75.4
	RLOR (Ours)	✓	**46.2**	**64.0**	**44.7**	**47.3**	**78.1**

Table 2: Inference efficiency and behavior statistics. We compare average token usage, pivot rate, reasoning steps, and tool calls among same-backbone methods.

Method	Avg. Tokens	Pivot Rate	Avg. Steps	Tool Calls
Mono	58216.31	0%	12.34	11.61
Mixture-SFT	52128.05	26.12%	11.62	11.57
Stage-SFT	45593.42	43.75%	10.41	10.35
GSPO	45652.79	42.23%	10.79	10.69
RLOR (Ours)	**41730.06**	**46.04%**	**8.90**	**8.83**

where β controls the strength of prefix protection. This objective implements pivot-aware credit assignment: correct trajectories without pivot are reinforced globally, correct trajectories with pivot are reinforced from the pivot onward, and incorrect trajectories are discouraged only after the estimated deviation boundary.

3 Experiment

3.1 Experimental Setup

We benchmark RLOR against three families of external reference systems together with a set of the same-backbone baselines. *Direct-inference models* answer without tool use and thus isolate the contribution of agentic interaction: GPT-4.1 (OpenAI, 2025a) and o4-mini (OpenAI, 2025e) from the OpenAI series, and the reasoning-oriented DeepSeek-R1-0528 (Guo et al., 2025b). *Proprietary agents* represent the strongest closed-source deep-research systems, including OpenAI DeepResearch (OpenAI, 2025c), Kimi-Researcher (Moonshot AI, 2025), GPT-5 (OpenAI, 2025b), and Gemini-3.0-Pro (Gemini Team, Google, 2026). *Open-source agents* target long-horizon information-seeking task family and provide open-weight references: ASearcher-Web (Gao et al., 2025), Tongyi DeepResearch (Tongyi DeepResearch Team, 2025), and the test-time-scaled GLM-4.5 Heavy (GLM-4.5 Team et al., 2025). Finally, for a controlled comparison under an identical backbone and tool interface, all *same-backbone baselines* are built on Qwen3.5-35B-A3B (Qwen Team, 2026): Mono is the ReAct-style base agent, Mixture-SFT mixes data from all three curriculum stages, Stage-SFT follows the staged curriculum of Section 2.3, and GSPO (Zheng et al., 2025) applies group sequence policy optimization without credit assignment.

We also report Best-of-N ($N \in \{1, \ldots, 5\}$) accuracy to evaluate performance under different inference budgets. Beyond accuracy, we characterize inference behavior using average generated tokens, pivot rate, average reasoning steps, and tool calls.

3.2 Main Results

Table 1 presents the main results on five long-horizon agent benchmarks. RLOR consistently

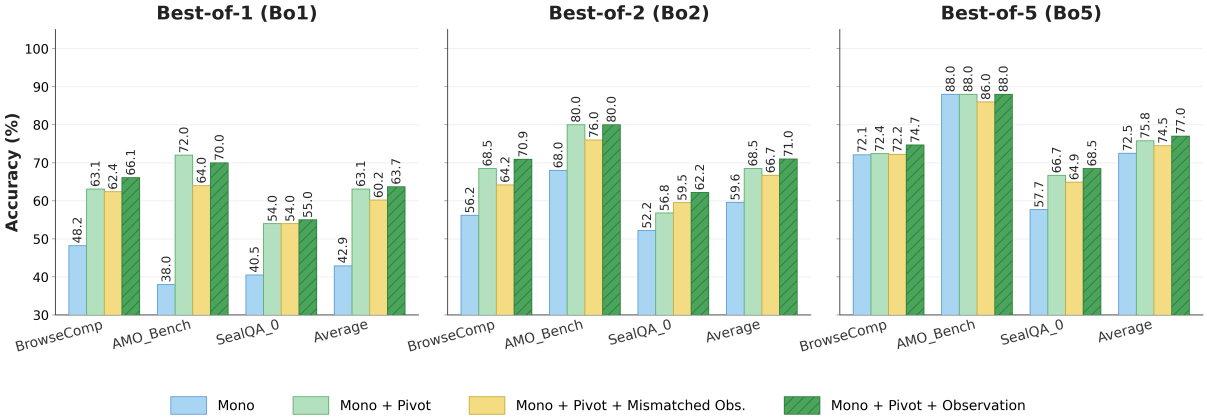

Figure 3: Ablation analysis of pivot and observation interventions on DeepSeek-V3.2-Thinking. Across Best-of-N settings, pivot-only intervention substantially improves over the original model. Matched observations further strengthen pivot-guided repair, whereas mismatched observations reduce the gains, indicating that effective trajectory repair depends on both accurate pivots and high-quality observations.

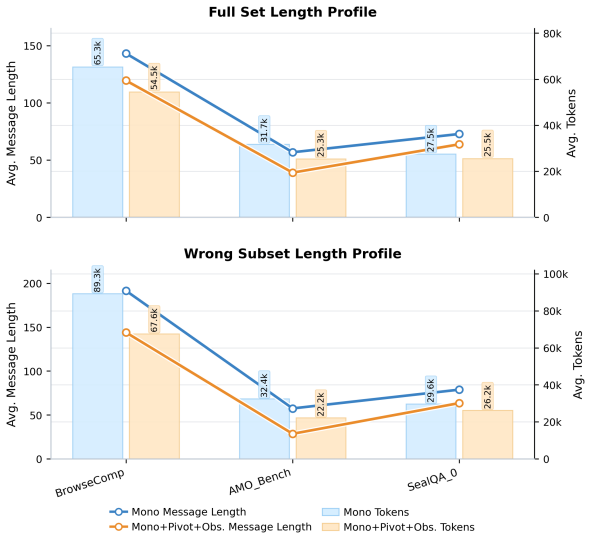

Figure 4: Efficiency analysis of pivot and observation interventions on DeepSeek-V3.2-Thinking. The top panel shows results for the full evaluation sets, and the bottom shows results for the subset of questions that the original model initially answered incorrectly. Bars indicate average token usage, and lines indicate the average number of messages per trajectory.

outperforms all same-backbone baselines across BrowseComp, AMO_Bench, SealQA_0, HLE, and GAIA. Compared with the strongest RL baseline, GSPO, RLOR achieves an average improvement of **2.8 points**, with particularly notable gains on AMO_Bench and SealQA_0. It also substantially improves over SFT-based variants, suggesting that RL optimization further enhances the model's ability to exploit pivot-localized repairs. Moreover, RLOR remains competitive with proprietary and open-source agent systems across the evaluated benchmarks, despite using Qwen3.5-35B-A3B as the backbone.

Table 2 further analyzes inference efficiency and behavior among same-backbone methods. RLOR achieves the lowest average token usage, reasoning steps, and tool calls, while maintaining the highest pivot rate. These results indicate that RLOR's performance gains do not stem from longer reasoning traces or heavier tool use. Instead, the higher pivot rate and lower inference cost suggest that RLOR learns to more effectively identify critical deviation points and perform localized repair, leading to more accurate and efficient long-horizon reasoning.

3.3 Ablation Study

We ablate pivot localization and observation quality on DeepSeek-V3.2-Thinking (Figure 3). As depicted, introducing pivots consistently improves over the Mono across all listed Best-of-N settings, indicating that explicitly exposing deviation points provides effective guidance for long-horizon reasoning. Building on top of this, adding matched observations further improves performance. In contrast, mismatched observations still outperform Mono but consistently lag behind the matched pivot-observation setting, showing that the improvement is not merely due to additional context. Instead, effective trajectory repair depends on both accurate pivot localization and semantically aligned observations that provide corrective guidance for subsequent reasoning. Additional results are provided in Appendix B.2.

Figure 4 further shows that pivot-observation intervention improves not only accuracy but also inference efficiency. Compared with Mono, the

matched pivot-observation setting reduces average token usage and message length. This reduction is more pronounced on the subset of questions initially answered incorrectly by the original model, where Mono tends to continue along erroneous reasoning paths and accumulate longer trajectories. By contrast, pivot-observation intervention enables the model to identify and correct deviations earlier, leading to shorter reasoning paths with fewer tokens and messages. These results suggest that pivot-guided repair improves the compactness and effectiveness of long-horizon reasoning.

3.4 Curriculum SFT Analysis

Table 3 ablates Stage-SFT. The curriculum progressively introduces pivot localization, pivot-rationale modeling, and repaired continuation, and achieves the best Bo1 performance among all SFT variants. Compared with Mixture-SFT, Stage-SFT improves average Bo1 accuracy by 8.7 points, indicating that mixing heterogeneous supervision signals is less effective. It also outperforms Only-Stage3, suggesting that directly imitating completely repaired trajectories is insufficient without first learning to identify pivots and diagnose their causes. The steady improvement from Stage1 to Stage3 further shows that pivot localization, rationale modeling, and continuation repair provide complementary supervision for long-horizon trajectory repair.

3.5 Accuracy–Cost Trade-off

We further analyze the accuracy–cost trade-off under token budgets from 8k to 128k, as shown in Figure 5. Under the same inference budget, RLOR consistently outperforms Mono and Stage-SFT, showing that it converts available computation into task accuracy more effectively. Moreover, when compared by realized token usage, RLOR reaches comparable or higher accuracy with fewer average tokens, forming a stronger Pareto frontier between accuracy and inference cost.

These results indicate that pivot-guided repair mechanism in RLOR encourages more targeted computation: the model preserves useful prefix reasoning, identifies consequential deviations, and allocates subsequent tokens to local trajectory correction rather than unproductive continuation.

3.6 Pivot usage and observation distribution

We analyze the pivot rate and observation-type distribution across benchmarks in Figure 6. The pivot rate varies substantially, reflecting the benchmark-

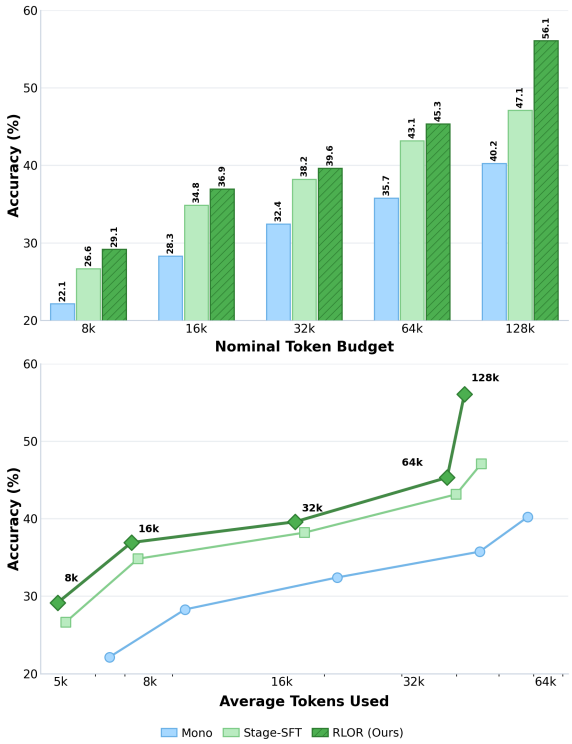

Figure 5: Analysis of inference performance under token-budget constraints. The top panel compares accuracy across nominal token budgets from 8k to 128k. The bottom panel visualizes the accuracy–cost relationship by plotting accuracy against the realized average token usage for each method.

specific intervention demand under the current policy. Higher pivot rates indicate that correct trajectories are more difficult for the model to maintain without localized repair, as observed on BrowseComp and SealQA. In contrast, GAIA requires far fewer pivot interventions.

The observation distribution further shows that RLOR adapts its repair behavior to different failure modes. Contextual pivots dominate on most benchmarks, while AMO_Bench contains a larger share of evaluative pivots, suggesting that its failures more often arise from unreliable intermediate reasoning. Overall, RLOR does not apply a fixed correction pattern, but learns to trigger local repair according to the structure of each benchmark.

4 Related Work

RLVR and Credit Assignment. Reinforcement learning with verifiable rewards (RLVR) has emerged as an effective paradigm for improving LLM reasoning in domains such as mathematics and code generation with checkable outcomes (Shao et al., 2024; Guo et al., 2025a). However, terminal rewards provide sparse supervision for long reasoning trajectories, making credit assignment

Table 3: Ablation study of curriculum design in Stage-SFT. We compare mixed-data SFT, only Stage-3-data SFT, and our staged curriculum. The curriculum progressively introduces pivot labels, pivot reasons, and repaired continuation, leading to the strongest overall performance across Best-of-N settings.

Method	BrowseComp					AMO_Bench					SealQA_0					Average		
	Bo1	Bo2	Bo3	Bo4	Bo5	Bo1	Bo2	Bo3	Bo4	Bo5	Bo1	Bo2	Bo3	Bo4	Bo5	Bo1	Bo3	Bo5
Base Model	32.7	42.2	46.9	48.1	51.5	32.0	56.0	64.0	68.0	70.0	27.1	34.3	44.2	46.9	49.6	30.6	51.7	57.0
Mixture-SFT	33.3	41.8	47.3	47.9	52.0	34.0	50.0	62.0	66.0	72.0	28.5	33.4	45.1	**50.5**	**52.3**	31.9	51.5	58.8
Only-Stage3	34.3	42.5	50.4	**52.6**	55.9	42.0	60.0	60.0	62.0	64.0	30.6	38.5	39.1	44.7	46.6	35.6	49.8	55.5
SFT-Stage1	34.1	42.1	50.7	52.1	**56.1**	44.0	60.0	62.0	64.0	72.0	29.8	36.0	36.7	37.8	42.3	36.0	49.8	56.8
SFT-Stage2	34.6	42.7	**50.9**	52.4	54.1	48.0	54.0	64.0	70.0	72.0	**33.2**	38.8	41.5	43.3	45.1	38.6	52.1	57.1
SFT-Stage3	**35.4**	**44.7**	47.3	52.0	53.3	**54.0**	**64.0**	**70.0**	**72.0**	**78.0**	32.5	**41.5**	**47.8**	47.8	50.5	**40.6**	**55.0**	**60.6**

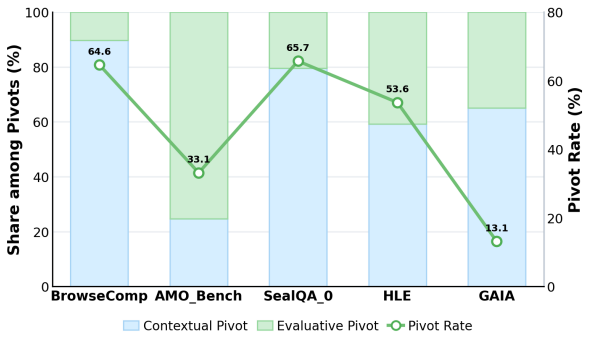

Figure 6: Pivot usage and observation-type distribution across benchmarks. Bars show the proportion of contextual and evaluative observations among triggered pivots. The line indicates the pivot rate on each benchmark.

difficult: incorrect rollouts may contain correct prefixes, while successful ones may include spurious or redundant steps. Recent studies mitigate this issue through finer-grained policy updates, including Monte Carlo token-level value estimation (Kazemnejad et al., 2024), outcome-grounded advantage reshaping (Li et al., 2026), and entropy-based analyses of critical decision tokens (Wang et al., 2026). While these methods highlight the importance of credit assignment, they mainly redistribute scalar rewards over existing trajectories weakening their capacity to diagnose the cause of failure or guide targeted repair.

Process Supervision and Critical Error Localization. Process supervision and process reward models reduce the sparsity of rewards by evaluating intermediate reasoning steps (Uesato et al., 2022; Lightman et al., 2024). Prior work constructs step-level supervision via human annotations, automatically generated signals, or search-based estimation, as in Math-Shepherd and OmegaPRM (Wang et al., 2024; Luo et al., 2024). Nevertheless, PRM-based approaches typically cast process information as evaluative scores or correctness labels, which may discard diagnostic details about why a trajectory deviates and how it should be redirected, limiting fine-grained, localized intervention.

Reasoning Pauses and Self-Correction. A related line of work improves reasoning by allocating additional computation or introducing feedback-driven revision. Pause-token methods insert learnable or dummy tokens to enable extra hidden-state computation before key outputs (Goyal et al., 2024), with later work studying dynamic pause insertion at low-confidence positions (Kim et al., 2025). Self-correction methods revise outputs through natural-language feedback, including iterative refinement, verbal memory, and tool-assisted critique (Madaan et al., 2023; Shinn et al., 2023; Gou et al., 2024). However, intrinsic self-correction remains unreliable without grounded feedback (Huang et al., 2024), and existing methods usually revise complete outputs or future trials. As a result, errors introduced at the point of deviation are allowed to propagate through the remainder of the trajectory before any correction is applied.

5 Conclusion

We proposed RLOR, an observation-guided trajectory repair framework for long-horizon reasoning. RLOR treats reasoning as an intervenable process by localizing deviations with pivots, injecting observation-based pivot reasons, and repairing only the suffix while preserving the valid prefix. Through curriculum SFT and pivot-aware RL, the model learns to solve directly, trigger pivots, and perform observation-conditioned repair. Experiments show that RLOR improves both accuracy and token efficiency, while ablations verify the roles of pivot localization, observation content, curriculum learning, and suffix-localized credit assignment. These results highlight our pivot-based localized trajectory repair as an effective mechanism for building more reliable and efficient long-horizon reasoning agents.

Limitations

This work has several limitations. First, our evaluation focuses on long-horizon reasoning benchmarks involving multi-step question answering and tool-augmented problem solving. Although these settings cover representative agentic reasoning scenarios, the effectiveness of RLOR in other domains, such as scientific discovery, codebase-level software engineering, or interactive planning, remains to be further examined.

Second, RLOR relies on teacher-annotated pivots and observation-based pivot reasons during curriculum supervised fine-tuning. As a result, the quality of the learned repair behavior may be affected by the accuracy and consistency of these annotations.

Third, while RLOR improves inference efficiency by repairing only the suffix after a pivot, its training pipeline introduces additional costs from trajectory annotation, staged supervised fine-tuning, and reinforcement learning. These costs may limit direct application in settings with very limited training resources.

References

Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. 2025. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous rl. In *NeurIPS 2025 Workshop on Multi-Turn Interactions in Large Language Models (MTI-LLM)*. Spotlight.

Gemini Team, Google. 2026. Gemini 3 Pro. https://deepmind.google/models/gemini/. Technical report / model release.

GLM-4.5 Team, Aohan Zeng, and 1 others. 2025. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, and 1 others. 2024. Critic: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, volume 2024, pages 57734–57811.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. 2024. Think before you speak: Training language models with pause tokens. In *International Conference on Learning Representations*, volume 2024, pages 27896–27923.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, and 1 others. 2025b. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *International conference on learning representations*, volume 2024, pages 32808–32824.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. 2024. Vineppo: Refining credit assignment in rl training of llms. *arXiv preprint arXiv:2410.01679*.

Eunki Kim, Sangryul Kim, and James Thorne. 2025. Learning to insert [pause] tokens for better reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 23760–23777.

Ziheng Li, Liu Kang, Feng Xiao, Luxi Xing, Qingyi Si, Zhuoran Li, Weikang Gong, Deqing Yang, Yanghua Xiao, and Hongcheng Guo. 2026. Outcome-grounded advantage reshaping for fine-grained credit assignment in mathematical reasoning. *arXiv preprint arXiv:2601.07408*.

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

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, and 1 others. 2024b. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pages 52989–53046.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, and 1 others. 2024. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594.

Moonshot AI. 2025. Kimi-Researcher: End-to-End RL Training for Emerging Agentic Capabilities. https://moonshotai.github.io/Kimi-Researcher/.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, and 1 others. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

OpenAI. 2025a. GPT-4.1. https://openai.com/index/gpt-4-1/. Technical report / model release.

OpenAI. 2025b. GPT-5 System Card. https://openai.com/index/gpt-5-system-card/. System card.

OpenAI. 2025c. Introducing Deep Research. https://openai.com/index/introducing-deep-research/.

OpenAI. 2025d. Introducing GPT-5.2. https://openai.com/index/introducing-gpt-5-2/. Accessed: 2026-05-26.

OpenAI. 2025e. OpenAI o3 and o4-mini System Card. https://openai.com/index/o3-o4-mini-system-card/. System card.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2024. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*, volume 2024, pages 9695–9717.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5.

Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2023. Training language models with language feedback at scale. *arXiv preprint arXiv:2303.16755*.

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. *Advances in neural information processing systems*, 36:68539–68551.

Zhihong Shao, Yuxiang Luo, Chengda Lu, ZZ Ren, Jiewen Hu, Tian Ye, Zhibin Gou, Shirong Ma, and Xiaokang Zhang. 2025. Deepseekmath-v2: Towards self-verifiable mathematical reasoning. *arXiv preprint arXiv:2511.22570*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.

Tongyi DeepResearch Team. 2025. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*.

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.

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

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.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, and 1 others. 2026. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Advances in Neural Information Processing Systems*, 38:115452–115486.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

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

A Implementation Details

The pre-trained policy model of Qwen3.5-35B-A3B can be downloaded via https://huggingface.co/Qwen/Qwen3.5-35B-A3B. The evaluation datasets of BrowseComp can be downloaded on https://openai.com/index/browsecomp/, AMO_Bench can be downloaded on https://huggingface.co/datasets/meituan-longcat/AMO-Bench, SealQA_0 can be downloaded on https://huggingface.co/datasets/vtllms/sealqa, HLE can be downloaded on https://huggingface.co/datasets/cais/hle, and GAIA can be downloaded on https://huggingface.co/datasets/gaia-benchmark/GAIA.

A.1 RL Training Details

We conduct RL training in a multi-turn agent setting. For each prompt, the policy samples multiple on-policy rollouts, enabling exploration over long-horizon reasoning trajectories and tool-augmented continuations. To accommodate long-context reasoning, we use a trajectory-level token budget of approximately 128K tokens, with separate truncation limits for prompts, model responses, and tool outputs. Unless otherwise specified, generation is performed with temperature 1.0 and top-p 1.0 during rollout collection. The main RL hyperparameters are summarized in Table 4.

A.2 SFT Training Details

We perform supervised fine-tuning (SFT) to train the model with pivot-aware trajectory supervision. To support long-context trajectories, we set the sequence cutoff length to 131,072 tokens. Training is conducted for 8 epochs with a learning rate of 1.0×10^{-5}, cosine learning-rate scheduling, and a warmup ratio of 0.05. We use BF16 precision and Megatron-style parallelism to train the MoE model efficiently. The main SFT hyperparameters are summarized in Table 5.

B Additional Results

B.1 Additional Best-of-N Results under the Same Backbone

We provide additional Best-of-N results in Figure 7 to complement the main same-backbone comparison. All methods in this analysis use Qwen3.5-35B-A3B as the base model and share the same inference-time tool interface. This controlled setting isolates the effect of the training strategy from differences in backbone capacity or tool access. We compare Mono, Mixture-SFT, Stage-SFT, GSPO, and RLOR, and report Bo1, Bo3, and Bo5 accuracy on BrowseComp, AMO_Bench, SealQA_0, GAIA, and HLE.

As shown in Figure 7, increasing the sampling budget generally improves performance across methods, indicating that these long-horizon benchmarks benefit from inference-time scaling. RLOR achieves consistent gains over the Mono and SFT-based variants across most benchmarks and sampling budgets, and remains competitive with the GSPO baseline under larger Best-of-N settings. The improvements are especially evident on BrowseComp, AMO_Bench, and SealQA_0, where pivot-aware trajectory repair provides stronger candidate trajectories as the sampling budget increases. Overall, the results show that the benefit of RLOR is not restricted to Bo1 inference; its pivot-aware training also improves or preserves performance under increased inference-time sampling.

B.2 Extended Best-of N Results on Pivot Localization and Observation Quality Ablation

To complement Figure 3, we extend the Pivot Localization and Observation Quality ablation on DeepSeek-V3.2-Thinking to a broader range of Best-of-N budgets. The setup is identical to Figure 3, but here we additionally report Bo3, Bo4, and the average across all Best-of-N settings for a more continuous view of how each intervention scales with the sampling budget.

As shown in Figure 8, the trend in Figure 3 holds across the expanded regime. Pivots consistently improve over Mono at every Best-of-N level, and matched observations achieve the highest accuracy across Bo1–Bo5 as well as their average. Mismatched observations again outperform Mono but stay clearly below the matched setting, confirming that the gain is not driven by additional context alone. Effective trajectory repair requires both accurate pivot localization and semantically aligned observations, and the averaged Best-of-N curve shows the matched–mismatched gap slightly widening with larger N, suggesting that observation quality becomes more valuable as more candidate trajectories are sampled.

Table 4: Main hyperparameters used for RL training and rollout collection. Maximum response length and maximum assistant turns are hard truncation limits rather than target generation lengths.

Hyperparameter	Value	Description
Training batch size	64	Prompts per RL training batch.
Validation batch size	256	Prompts per validation batch.
Optimization mini-batch size	32	Mini-batch size for policy optimization.
Rollouts per prompt	8	On-policy trajectories sampled per prompt.
Validation samples per prompt	3	Sampled responses used for validation.
Temperature / top-p	1.0 / 1.0	Sampling configuration for rollout collection.
Maximum prompt length	10,240	Token limit for input prompts.
Maximum response length	120,832	Token limit for model-generated responses.
Maximum tool response length	32,768	Token limit for tool outputs.
Maximum assistant turns	256	Hard truncation limit on assistant turns.
vLLM maximum sequences	64	Maximum sequences processed by the rollout engine.
vLLM maximum batched tokens	131,072	Maximum tokens batched by the rollout engine.
Parallel tool calls	3	Maximum concurrent tool calls per agent step.
Agent workers	8	Workers for parallel agent execution.

Table 5: Main hyperparameters used for supervised fine-tuning. We report the optimization, sequence-length, and distributed-training settings most relevant to reproducing the SFT stage.

Hyperparameter	Value	Description
Fine-tuning type	Full	All language model parameters are updated.
Chat template	qwen3_5	Conversation template used for data formatting.
Cutoff length	131,072	Maximum sequence length used during SFT.
Per-device batch size	1	Number of examples processed per device.
Gradient accumulation steps	2	Number of steps accumulated before each optimizer update.
Training epochs	8	Number of passes over the SFT dataset.
Learning rate	1.0×10^{-5}	Peak learning rate for full-parameter SFT.
LR scheduler	Cosine	Learning-rate decay schedule.
Warmup ratio	0.05	Fraction of training used for learning-rate warmup.
Precision	BF16	Numerical precision used during training.
Tensor parallel size	2	Number of partitions for tensor parallelism.
Expert parallel size	4	Number of partitions for MoE expert parallelism.
Sequence parallelism	Enabled	Sequence-parallel optimization for distributed training.
Pipeline parallel size	1	No pipeline partitioning is used.
Context parallel size	1	No additional context parallelism is used.
Distributed optimizer	Enabled	Distributed optimizer for memory-efficient training.
Optimizer CPU offload	Enabled	Optimizer states are offloaded to CPU memory.
MoE token dispatcher	All-to-all	Dispatcher used for routing tokens across MoE experts.
MoE grouped GEMM	Enabled	Grouped GEMM optimization for MoE expert computation.
Activation recomputation	Full	Full recomputation to reduce activation memory.

C Prompt-based Annotation for Curriculum SFT

We construct the curriculum SFT data using a teacher-based trajectory annotation pipeline. Given a raw reasoning trajectory, the teacher first acts as a reasoning trajectory auditor and identifies the earliest assistant step that deviates from a valid reasoning path. This annotation provides the pivot supervision used in the first curriculum stage and the pivot-reason supervision used in the second stage. To make the annotation format deterministic and easy to parse, we require the teacher to return a strict JSON object containing the pivot step index, the corresponding assistant message index, and a short natural-language pivot reason.

The auditor prompt used for pivot annotation is shown below:

Prompt for Stage 1 and 2 in Curriculum-SFT

```
You are an expert reasoning trajectory auditor.
Given a trajectory, identify the earliest assistant
      reasoning step that starts to deviate from a
      valid path.
Return strict JSON with only these fields:
    - pivot_step_index: integer >= 1
    - pivot_message_index: integer message index of
      the pivot assistant step
    - pivot_reason: short string
Rules:
    - Pick the earliest harmful assistant step.
    - If uncertain, still choose the earliest
      plausible error point.
    - Output JSON only.
```

Figure 7: Best-of-N accuracy under the same-backbone setting. All methods use Qwen3.5-35B-A3B as the base model and are evaluated with the same inference-time tool interface. Results are reported for Bo1, Bo3, and Bo5 on BrowseComp, AMO_Bench, SealQA_0, GAIA, and HLE.

For the third curriculum stage, we further construct pivot-conditioned continuation examples. Specifically, we first reuse the auditor prompt to identify the pivot boundary and obtain the corresponding pivot reason. We then preserve the trajectory prefix before the pivot and append the pivot intervention as process-level feedback. Conditioned on this repaired state, the teacher generates candidate continuations, from which we retain trajectories that correctly solve the original task. This produces supervision for continuation repair, where the model learns to resume reasoning from the localized intervention rather than rewriting the entire trajectory.

Generation Prompt for Stage 3 in Curriculum-SFT

```
You are an expert reasoning assistant.
You are given a reasoning trajectory that was
      interrupted by a pivot intervention.
The pivot reason explains the error made. Your task is
       to continue the reasoning from the pivot point,
      addressing the identified issue and arriving at
      the correct answer.
Rules:
    - Do NOT repeat the previous erroneous reasoning.
    - Address the pivot reason directly and concisely.
    - Continue to reason step by step until you reach
      a final answer.
    - End your response with:
      Reasoning: <brief summary of key findings>
      Answer: <final answer>
    - Output plain text only (no JSON, no markdown
      code fences).
```

Judge Prompt for Expert-Label Annotation

```
You are a strict trajectory quality judge.
Evaluate continuation quality after a pivot
      intervention.
Return STRICT JSON with fields:
    - verify_pass: boolean (whether final result
      appears correct for the question)
    - alignment_pass: boolean (whether continuation
      clearly responds to pivot reason)
    - verify_reason: short string
    - alignment_reason: short string
```

Overall, the annotation pipeline decomposes trajectory repair into three progressively harder learning targets: locating the earliest deviation, explaining the reason for intervention, and generating a corrected suffix from the preserved prefix and pivot reason. This design is consistent with our curriculum SFT formulation, where the model first learns pivot identification, then pivot-reason generation, and finally observation-conditioned repair.

D Case Study: Pivot-based Correction of Trajectory Drift

This appendix presents a representative case in which a pivot intervention corrects an early trajectory drift. The task requires the model to identify the minute of a match-winning goal and the corresponding football tournament from several indirect constraints, including a player's birthplace, a nineteenth-century fort near a port, shared club histories, and the distance between the match stadium and a monument.

Gold answer. The gold answer for this case is:

`109, EURO 2016.`

Original trajectory. The original trajectory initially decomposes the question into several constraints: identifying an opposing player born in a port city with a nineteenth-century fort, checking whether this player shared club history with players from the scorer's team, and matching the stadium to a nearby monument. However, after an initial broad search, the model makes an unsupported transition to *Fort Sumter* and *Charleston*. This candidate is not grounded in the preceding search results and does not satisfy the stated constraint that the fort should be approximately 400–600 meters from the city's port.

After this unsupported entity assumption, the reasoning trajectory drifts further away from the target solution. The model eventually constructs an unrelated hypothesis involving a Bundesliga match and outputs:

`32nd minute, Bundesliga.`

This answer is inconsistent with the gold answer and is therefore incorrect.

Pivot intervention. The pivot intervention is applied at the first major point of divergence. Instead of correcting only the final answer, the intervention targets the unsupported reasoning transition itself. The intervention observes that the move from a generic clue about a nineteenth-century fort near a port to *Fort Sumter/Charleston* is not justified by the available evidence and likely initiates the trajectory drift. The model is then instructed to continue reasoning from this corrected point.

Repaired trajectory. After the pivot intervention, the model abandons the unsupported *Fort Sumter/Charleston* branch and resumes searching under the original constraints. The repaired trajectory eventually converges to the UEFA Euro 2016 final between Portugal and France. In that match, Portugal's Éder scored the decisive goal in extra time at the 109th minute. The repaired final answer is:

`109th minute, UEFA Euro 2016.`

This matches the gold answer.

Table 6 summarizes the key differences between the original and repaired trajectories.

Aspect	Original trajectory	Pivot-repaired trajectory
Initial reasoning	Decomposes the question into constraints involving fort, port, club history, stadium, and monument.	Starts from the same fixed prefix and preserves the original task decomposition.
First major drift	Introduces *Fort Sumter/Charleston* without sufficient evidence from the search results.	Identifies the *Fort Sumter/Charleston* move as unsupported and rejects that branch.
Failure mode	The unsupported entity assumption redirects subsequent search and reasoning toward an unrelated match hypothesis.	The trajectory is redirected toward candidates that better satisfy the original constraints.
Final hypothesis	Bundesliga match hypothesis.	UEFA Euro 2016 final, Portugal vs. France.
Final answer	`32nd minute, Bundesliga.`	`109th minute, UEFA Euro 2016.`
Correctness	Incorrect.	Correct.

Table 6: Comparison between the original trajectory and the pivot-repaired trajectory. The pivot intervention corrects an early unsupported entity-level assumption, preventing the trajectory from continuing along an irrelevant search branch.

This case illustrates a typical trajectory drift pattern: an early unsupported assumption introduces an incorrect entity, and subsequent reasoning attempts to rationalize that entity rather than re-evaluating the original constraints. The pivot intervention is effective because it targets the first unsupported transition rather than merely replacing the final answer. Once this transition is corrected, the model no longer follows the unrelated Bundesliga branch and instead recovers the correct final answer.

This case should be interpreted as an example of *answer-level recovery through pivot-based trajectory correction*, rather than as evidence that every intermediate step in the repaired trajectory is perfectly verified. The repaired trajectory reaches the correct answer, but some intermediate clue alignments remain partially heuristic. Therefore, the case is most appropriate for demonstrating that a pivot can prevent an early unsupported assumption from dominating the final answer.

E vLLM server

During testing, we deploy the trained model using vLLM. The model service is exposed through a stan-

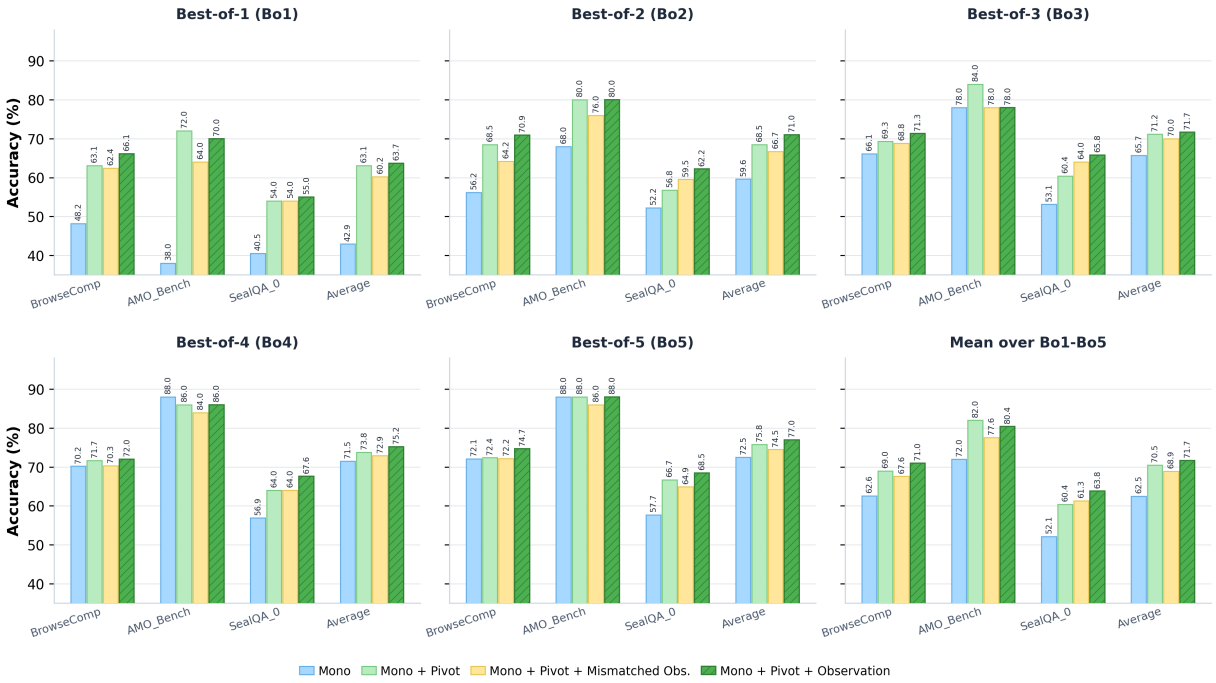

Figure 8: Ablation analysis of pivot and observation interventions on DeepSeek-V3.2-Thinking. Across Best-of-N settings, pivot-only intervention substantially improves over the original model. Matched observations further strengthen pivot-guided repair, whereas mismatched observations reduce the gains, indicating that effective trajectory repair depends on both accurate pivots and high-quality observations.

dardized HTTP/OpenAI-compatible interface, supports long-context inference, and can be efficiently scaled in a multi-GPU environment. For each evaluation round, the client maps the constructed query set into requests and sends them to the vLLM server. After receiving the responses, the client writes them into the interaction history, thereby supporting the parallel inference pipeline from information acquisition to final output decision.

vLLM Serving Command

```
vllm serve ${MODEL_PATH} \
    --port ${PORT} \
    --host :: \
    --gpu-memory-utilization 0.8 \
    --max-model-len 32768 \
    --tensor-parallel-size 8 \
    --served-model-name expert \
    --trust-remote-code \
    --disable-log-requests
```