LEARNING TO LOCALIZE AND REPAIR DEVIATIONS IN LONG-HORIZON REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) agents increasingly tackle complex long-horizon tasks through multi-step reasoning, tool use, and environment interaction. However, local errors can propagate through subsequent steps, derailing reasoning trajectories and wasting computation. Existing approaches often provide scalar supervision with limited corrective guidance or rely on reflection and revision that incur additional inference overhead. We therefore propose **RLOR**, an observation-guided trajectory repair framework. RLOR identifies the earliest deviation as a *pivot*, injects observation-based natural-language feedback, and repairs only the subsequent continuation while preserving the valid prefix. We use multi-round pivot annotations to guide curriculum supervised fine-tuning (SFT) and pivot-aware reinforcement learning (RL) with suffix-localized credit assignment. This training enables agents to internalize and generalize repair behavior, allowing autonomous correction without external teacher feedback at deployment. Experiments on long-horizon reasoning benchmarks show that RLOR improves accuracy and token efficiency, with ablations confirming the contributions of pivot localization, observation content, and the training design.

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., 2023; 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 tasks such as multi-hop question answering and tool-augmented problem solving, agents must maintain task goals, integrate external evidence, and adapt their actions across multiple steps. A central challenge is that local errors can propagate: an unsupported hypothesis or overlooked constraint can mislead subsequent reasoning and tool use, despite useful progress made earlier in the trajectory. Consequently, simply extending the context window or allocating additional inference steps does not ensure reliable use of accumulated information. Instead, it may prolong exploration within an invalid search space (Liu et al., 2024a; Huang et al., 2024). Reliable long-horizon reasoning therefore requires mechanisms for recovering from local deviations while retaining useful progress.

Prior work has explored finer-grained supervision and feedback to improve reasoning. Process-supervision and verifier-based approaches provide denser signals for evaluating reasoning than final-answer supervision(Lightman et al., 2024; Wang et al., 2024; Shao et al., 2025). However, when expressed as correctness labels or scalar rewards, these signals offer limited explicit guidance on why a trajectory deviates and how to redirect it. Natural-language feedback and self-refinement methods provide richer corrective signals through critiques, reflections, or tool-grounded validation (Madaan et al., 2023; Shinn et al., 2023; Gou et al., 2024). Yet these methods primarily rely on feedback-and-revision loops that introduce additional inference overhead at deployment through extra token generation or tool calls. Our goal is therefore to learn targeted trajectory repair as part of the policy, enabling correction within the ongoing rollout during inference.

To this end, we propose **RLOR**, an observation-guided trajectory repair framework for long-horizon reasoning, enabling models to internalize local correction with natural language. Specifically, it identifies the earliest step at which a trajectory departs from a valid reasoning path, termed the *pivot*. At this point, it introduces an observation-based *pivot reason* that restates unsupported hypothesis or

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

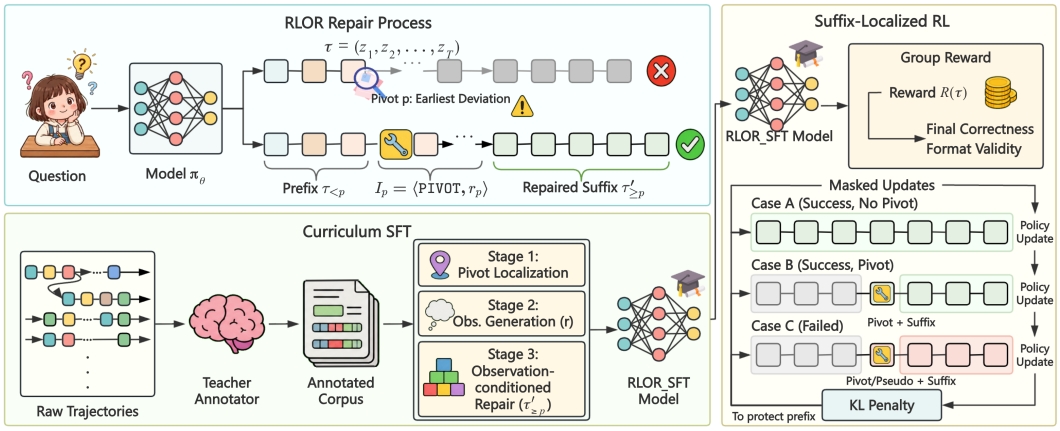


Figure 1: Overview of RLOR and its training pipeline. RLOR repairs trajectory deviations in long-horizon reasoning by localizing the earliest deviation as a pivot, injecting an observation-based pivot reason, preserving the valid prefix, and generating a repaired suffix. The model is trained through 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.

overlooked task constraints. The model preserves the valid prefix preceding the pivot and generates a repaired suffix conditioned on this prefix and the feedback, reusing established reasoning rather than restarting from scratch. Through training, the model learns to autonomously localize deviations, generate pivot reasons, and repair subsequent reasoning, enabling local correction without external teacher feedback at deployment.

Long-horizon reasoning may involve multiple deviations, with new errors arising even after earlier ones have been corrected. To address successive deviations, we extend pivot annotation and trajectory repair to multiple rounds during data construction. For trajectories that remain incorrect after a repair, the next annotation round identifies the earliest new deviation in the repaired continuation and repeats the prefix-preserving intervention. The resulting supervision captures successive local corrections rather than only a single revision of the original rollout.

To internalize this behavior, curriculum supervised fine-tuning (SFT) sequentially teaches pivot localization, observation-based reason generation, and conditioned suffix repair. Pivot-aware reinforcement learning (RL) then uses trajectory-level outcomes to further optimize direct solving and local repair. After this optimization, the policy can triggers pivots and generates its own feedback without requiring an external teacher during inference.

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, showing the effectiveness of observation-guided reasoning repair. Further ablations analyze the roles of pivot localization, observation content and training design. 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 prefix while correcting unreliable trajectory segments.

Our contributions are summarized as follows:

- We formulate long-horizon reasoning as an intervenable proces, coupling deviation localization, observation-based feedback, and suffix repair while preserving the valid prefix.
- We propose RLOR, which combines multi-round repair supervision, curriculum SFT, and pivot-aware RL with suffix-localized credit assignment to learn autonomous repair without external teacher feedback at deployment.
- We demonstrate improved accuracy and token efficiency on long-horizon reasoning benchmarks, with ablations examining the contributions of the pivot localization, observation content and training design.

2 RLOR: REINFORCEMENT LEARNING FOR OBSERVATION-GUIDED REASONING REPAIR

We propose RLOR (Figure 1), an observation-guided training framework that improves long-horizon reasoning through localized trajectory repair. RLOR identifies the earliest deviation as a pivot, and injects observation-based natural-language feedback 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. The same intervention can be applied repeatedly to newly generated continuations to address successive deviations. We internalize this behavior into policy model through curriculum supervised fine-tuning, and pivot-aware reinforcement learning.

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 has no pivot.

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 model then generates a repaired suffix:

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

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. By inserting the pivot and its associated reason at the point of deviation, RLOR reshapes the distribution of subsequent reasoning, steering the trajectory back while preserving the established prefix.

Multi-round repair. Even after a local deviation is corrected, new errors may emerge as long-horizon reasoning progresses. We therefore extend the same repair operator to multiple rounds. Let $\tau^{(0)}$ denote the original trajectory. At round m, a pivot $p^{(m)}$ is localized in the current trajectory $\tau^{(m-1)}$, and repair proceeds as

$$h^{(m)} = \tau^{(m-1)}_{<p^{(m)}}, \qquad s^{(m)} \sim \pi_\theta\Big(\cdot \mid x, h^{(m)}, I^{(m)}\Big), \qquad \tau^{(m)} = \Big(h^{(m)}, I^{(m)}, s^{(m)}\Big).$$

where $h^{(m)}$ denotes the retained prefix before the current pivot, $I^{(m)} = \langle \texttt{PIVOT}, r^{(m)} \rangle$ the corresponding intervention, and $s^{(m)}$ the newly generated repaired suffix.

During offline annotation, only trajectories that remain incorrect are revisited, and subsequent pivots are restricted to the continuation generated after the preceding ones. Each round thus retains earlier interventions and newly established valid progress while replacing only the continuation following the current deviation. The process terminates upon success or exhaustion of a predefined repair-round budget. Single-round repair is the special case with a budget of one.

2.3 CURRICULUM SFT FROM MULTI-ROUND REPAIR SUPERVISION

We construct teacher-annotated repair supervision; annotation consistency is evaluated in Section 3.3. Given an initial trajectory collection $\mathcal{T} = \{(x_i, \tau_i^{(0)})\}_{i=1}^{N}$, we use GPT-5.2 (OpenAI, 2025d) to identify the earliest deviation $\hat{p}_i^{(m)}$ in the current trajectory at each round and generate its observation-based reason $\hat{r}_i^{(m)}$. A new continuation is then generated using the intervention procedure above, yielding $\tau_i^{(m)}$. Aggregating annotations across rounds gives

$$\mathcal{D}_{\text{ann}} = \bigcup_{i=1}^{N} \left\{ \left(x_i, \tau_i^{(m-1)}, \hat{p}_i^{(m)}, \hat{r}_i^{(m)}, \tau_i^{(m)} \right) \right\}_{m=1}^{M_i},$$

where M_i is the number of repair rounds performed for trajectory i.

We organize these annotations into three sequential learning tasks. The first stage teaches *pivot localization*, identifying the boundary at which local intervention is needed. The second teaches *pivot-reason generation*, producing a pivot marker together with an evaluative or contextual observation. The third teaches *conditioned continuation repair*, generating a repaired suffix from the retained prefix and intervention package. For the third stage, only continuations that pass the final-answer correctness check are retained, rather than directly imitating unsuccessful repair attempts.

Multi-round supervision preserves this curriculum order while broadening the reasoning states encountered at each stage. Later-round examples contain retained interventions and reasoning progress from previous rounds, teaching the model to localize, explain, and correct further deviations in contexts that have already undergone repair.

For curriculum stage $\ell \in \{1, 2, 3\}$, we optimize

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

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

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

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

Trajectory-specific update masks. Let $\mathcal{P}_i = \{p_{i,1} < \cdots < p_{i,n_i}\}$ denote the positions of generated pivot markers in τ_i. We partition output tokens into a main policy-update set Ω_i and a weakly KL-regularized set Ψ_i. Accordingly, we apply update masks based on whether a trajectory contains a pivot and whether it reaches a correct answer. Examples are provided in Appendix A.3.

Direct-Solve: Correct, no pivot. RLOR reinforces the entire correct no-pivot assistant trajectory: $\Omega_i = \text{Tokens}(\tau_i), \Psi_i = \emptyset$.

Localized repair: correct, with pivots. The main objective covers the first pivot marker, its feedback, and the full subsequent continuation: $\Omega_i = \text{Tokens}(\tau_{i,\geq p_{i,1}}), \Psi_i = \text{Tokens}(\tau_{i,<p_{i,1}})$.

Localized-Suppression (incorrect). If pivots are present, we use the first pivot as the update boundary. Otherwise, we introduce a pseudo-pivot $\tilde{p}_i$ at the assistant step with the highest policy entropy under the old policy, motivated by Wang et al. (2025). We define $b_i = \begin{cases} p_{i,1}, & n_i > 0, \\ \tilde{p}_i, & n_i = 0, \end{cases}$ The update mask is then set as follows: $\Omega_i = \mathrm{Tokens}(\tau_{i,\geq b_i}), \Psi_i = \mathrm{Tokens}(\tau_{i,<b_i})$. The pseudo-pivot only determines the update mask; it does not insert corrective feedback into rollout.

Credit assignment across multiple pivots. For a trajectory containing multiple pivots, the update set beginning at the first pivot includes all subsequent continuations. These tokens share the same terminal-reward advantage $\hat{A}_i$, without introducing separate rewards or local advantages for individual repair rounds. Prefix regularization therefore applies only before the first pivot.

Masked GSPO objective. We optimize Ω_i with a masked sequence-level GSPO (Zheng et al., 2025) objective. For token y_j with autoregressive context c_j, the length-normalized masked sequence-level ratio is defined as $\rho_i(\theta) = \exp\left[\frac{1}{|\Omega_i|}\sum_{j\in\Omega_i}\log\frac{\pi_\theta(y_j|c_{i,j})}{\pi_{\theta_{\mathrm{old}}}(y_j|c_{i,j})}\right]$. Using the clipped surrogate $g(\rho, A) = \min\left(\rho A,\ \mathrm{clip}(\rho, 1-\epsilon_{\mathrm{clip}}, 1+\epsilon_{\mathrm{clip}})A\right)$, the main objective is

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

To protect valid prefixes during RL updates, RLOR exclude Ψ_i from the main reward-driven objective and apply weak KL regularization against the old policy:

$$\mathcal{L}_{\mathrm{KL}} = \mathbb{E}_x\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{\max(1,|\Psi_i|)}\sum_{j\in\Psi_i} D_{\mathrm{KL}}\left(\pi_\theta(\cdot \mid c_{i,j}) \,\|\, \pi_{\theta_{\mathrm{old}}}(\cdot \mid c_{i,j})\right)\right].$$

The corresponding term is zero when $\Psi_i = \emptyset$. The final objective is $\max_\theta \quad \mathcal{J}_{\mathrm{RLOR}}(\theta) = \mathcal{J}_{\mathrm{main}}(\theta) - \beta\mathcal{L}_{\mathrm{KL}}$, where β controls the strength of prefix protection. This objective uses pivots to determine where reward-driven updates apply and group-relative advantages to determine direction, jointly optimizing direct solving and multi-round local repair without additional rewards.

3 Experiment

3.1 Experimental Setup

We evaluate RLOR against Rethink-workflow and other same-backbone baselines, with three families of literature-reported systems as references. *Direct-inference models* answer without tool use: GPT-4.1 (OpenAI, 2025b) and o4-mini (OpenAI, 2025e) from the OpenAI series, and the reasoning-oriented DeepSeek-R1-0528 (Guo et al., 2025). *Proprietary agents* represent the strongest closed-source deep-research systems, including OpenAI DeepResearch (OpenAI, 2025a), Kimi-Researcher (Moonshot AI, 2025), GPT-5 (OpenAI, 2025c), Gemini-3.0-Pro (Gemini Team, Google, 2026), and MiroThinker-H1 (MiroMind Team et al., 2026). *Open-source agents* target the long-horizon information-seeking task family and provide open-weight references: Tongyi DeepResearch (Tongyi DeepResearch Team, 2025), the test-time-scaled GLM-4.5 Heavy (GLM-4.5 Team et al., 2025), and QUEST-35B (Xie et al., 2026). *Rethink-workflow methods* interleave generation with explicit feedback-and-revision loops rather than learning a localized repair policy: Reflexion (Shinn et al., 2023), CRITIC (Gou et al., 2024), and Self-Refine(Madaan et al., 2023). The Rethink-workflow and RLOR blocks share Qwen3.5-35B-A3B (Qwen Team, 2026) and our evaluation harness. Within the RLOR block, 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) starts from the Stage-SFT checkpoint and applies group sequence policy optimization without RLOR's pivot-aware credit assignment. The **-Multi** suffix denotes the multi-round pivot-repair setting.

We evaluate RLOR on five long-horizon agentic benchmarks: BrowseComp (Wei et al., 2025), AMO-Bench (An et al., 2025), SealQA_0 (Pham et al., 2026), HLE (Phan et al., 2026), and GAIA (Mialon et al., 2024), which jointly assess reasoning under information seeking, evidence integration, and tool

Table 1: Main results on five long-horizon agent benchmarks. Rethink-Workflow and RLOR blocks use Qwen3.5-35B-A3B under the same evaluation harness. Direct Inference, Proprietary Agents, and Open-Source Agents contain scores from the original papers/technical reports, included for reference because their evaluation harnesses may differ. Tools indicates external tool use; **-Multi** denotes multi-round pivot repair. Bold/underline indicate best/second-best values within each block.

Type	Model	Tools	BrowseComp	AMO_Bench	SealQA_0	HLE	GAIA
Direct Inference	GPT-4.1	✗	1.5	4.1	–	4.9	<u>22.3</u>
	o4-mini	✗	**6.1**	**40.2**	<u>4.5</u>	**18.9**	**33.3**
	DeepSeek-R1-0528	✗	<u>2.0</u>	<u>34.3</u>	**5.4**	<u>17.7</u>	16.5
Proprietary Agents	OpenAI DeepResearch	✓	51.5	–	–	26.6	67.4
	Kimi-Researcher	✓	–	–	36.0	26.9	–
	GPT-5	✓	54.9	<u>52.4</u>	<u>51.4</u>	35.2	<u>76.4</u>
	Gemini-3.0-Pro	✓	<u>59.2</u>	**63.1**	45.5	<u>46.9</u>	–
	MiroThinker-H1	✓	**88.2**	–	**61.3**	**47.7**	**88.5**
Open-Source Agents	GLM-4.5 Heavy	✓	**54.0**	–	–	–	66.0
	Tongyi DeepResearch	✓	43.4	–	–	<u>32.9</u>	<u>70.9</u>
	QUEST-35B	✓	<u>45.4</u>	–	–	**37.2**	**80.8**
Rethink-Workflow	Reflexion	✓	**50.6**	<u>54.0</u>	**42.9**	**46.2**	**73.8**
	CRITIC	✓	<u>40.2</u>	50.0	<u>36.2</u>	<u>42.9</u>	<u>67.9</u>
	Self-Refine	✓	26.1	**56.0**	13.5	28.6	41.8
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	✓	35.4	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	<u>78.1</u>
	Mixture-SFT-Multi	✓	33.6	38.0	30.6	42.2	72.5
	Stage-SFT-Multi	✓	38.3	60.0	38.7	44.8	74.3
	GSPO-Multi	✓	<u>49.4</u>	<u>66.0</u>	<u>46.6</u>	<u>48.2</u>	76.9
	RLOR-Multi (Ours)	✓	**52.7**	**70.0**	**50.3**	**49.8**	**78.6**

interaction. For HLE and GAIA, we restrict our evaluation to their text-only subsets. 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 shows that RLOR consistently outperforms Mono, Mixture-SFT, Stage-SFT, and GSPO across all evaluated benchmarks under the same backbone and evaluation harness. These results support the effectiveness of the proposed training framework, with the gains over GSPO highlighting the added value of pivot-aware credit assignment. RLOR also outperforms the Rethink-workflow baselines on most benchmarks. RLOR-Multi further improves performance across all benchmarks and outperforms all methods using the same backbone and evaluation harness, indicating that the benefits of pivot-aware optimization persist under multi-round repair training paradigm.

Table 2 further analyzes inference efficiency and behavior among same-backbone methods. RLOR achieves the lowest average token usage, reasoning steps, and tool calls. 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 identifies critical deviations and repairs them locally, improving long-horizon accuracy and efficiency. Appendix B.2 provides additional efficiency comparisons with the Rethink-workflow baselines.

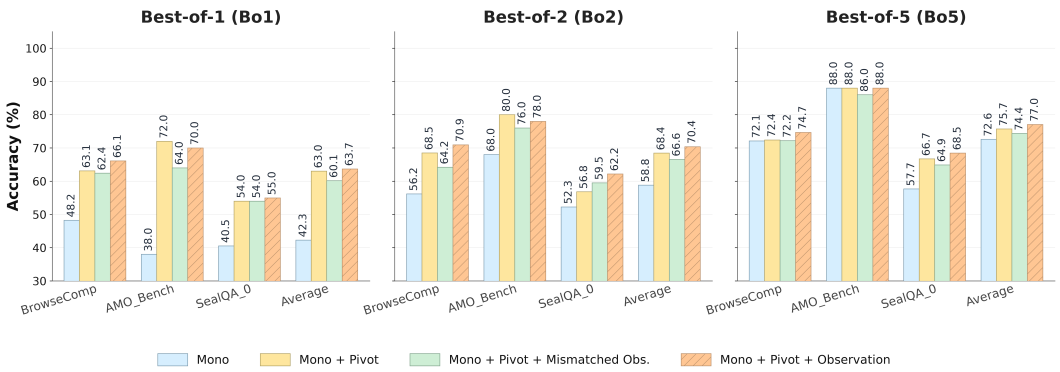


Figure 2: Intervention analysis on DeepSeek-V3.2-Thinking. Pivot-only intervention generally improves accuracy over Mono; matched observations generally outperform mismatched observations, supporting the usefulness of pivot information and semantically aligned feedback.

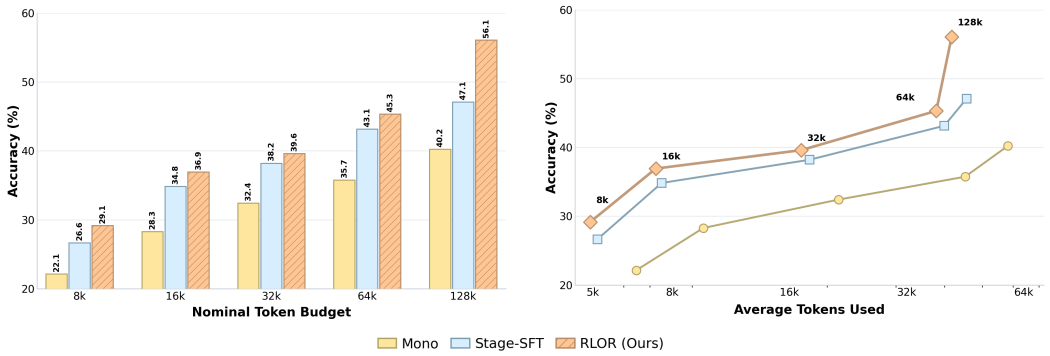


Figure 3: Analysis of performance under token-budget constraints. The left panel compares accuracy across nominal token budgets from 8k to 128k. The right panel visualizes the accuracy–cost relationship by plotting accuracy against the realized average token usage for each method.

3.3 Consistency and Utility of Pivot Supervision

Consistency of pivot annotations. We evaluate the stability and consistency of pivot annotations. Table 3 shows that pivot annotations are highly repeatable across 3 independent GPT-5.2 runs and remain reasonably consistent across different annotators. The comparison with human annotations further indicates substantial agreement in pivot locations and error categories, while some uncertainty remains in exact pivot boundaries and the semantic formulation of pivot reasons. Overall, both pivot locations and pivot reasons exhibit substantial consistency across repeated, cross-model, and human comparisons, supporting the stability and reliability of the annotation procedure. These results highlight the effectiveness of the pivot annotation design.

Utility of pivot information. We next test whether supplying pivot locations and observations improves DeepSeek-V3.2-Thinking through intervention ablations without training (Figure 2). Pivot-only intervention improves on Mono in most reported settings. Matched observations generally yield higher accuracy than mismatched ones, supporting the usefulness of semantically aligned pivot reason for guiding continuation repair. Extended ablations and multi-round teacher comparisons appear in Appendices B.3 and B.5, respectively. Matched pivot-observation interventions also reduce average token usage and messages per trajectory, especially on initially incorrect questions. Appendix B.3 provides more the efficiency analysis.

3.4 Multi-Round Curriculum SFT Analysis

Table 4 examines whether the gains of curriculum SFT come from the staged organization of repair supervision. Under the multi-round setting, the full three-stage curriculum achieves the strongest

Table 2: Inference efficiency and behavior under the same backbone, including average token usage, pivot rate, reasoning steps, and tool calls.

Method	Avg. Tokens	Pivot Rate	Avg. Steps	Tool Calls
Mono	58216	0%	12.34	11.61
Mixture-SFT	52128	26.12%	11.62	11.57
Stage-SFT	45593	43.75%	10.41	10.35
GSPO	45653	42.23%	10.79	10.69
RLOR (Ours)	**41730**	**46.04%**	**8.90**	**8.83**

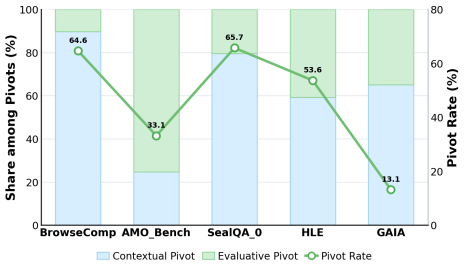


Figure 4: Pivot usage and observation-type distribution across benchmarks. Bars show observation proportions; the line shows the pivot rate.

Table 3: Pivot annotation consistency. Rates are percentages; mean distance is measured in assistant steps (lower is better). GPT-5.2 repeats use three independent runs; cross-model compares four annotators (GPT-5.2, DeepSeek-v4-Flash, Gemini-3.6-Flash and GLM-5.2). Additional metrics and definitions appear in Appendix B.6.

Comparison	Exact position	Within ± 1 step	Mean distance	Category agreement	Reason-semantic agreement
GPT-5.2 repeats	88.9	92.2	2.711	88.9	85.6
Cross-model	74.4	80.9	4.524	81.6	73.7
GPT-5.2 vs. human	76.7	90.4	3.833	90.2	73.3

overall performance across sampling budgets, outperforming mixed supervision and continuation-only training in aggregate. The progression from pivot localization to pivot-reason generation and repaired continuation also yields clear improvements under Bo1 across all three benchmarks. These results support progressively teaching where to intervene, why the trajectory deviates, and how to repair the continuation, rather than mixing all supervision or learning repair alone. Single-round and trajectory-source ablations are provided in Appendices B.4 and B.7.

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 3. Increasing the inference budget improves accuracy across all settings. Under the same inference budget, RLOR consistently outperforms Mono and Stage-SFT, showing that it converts available computation into task accuracy more effectively across the budget range. 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 4. The pivot rate varies substantially, reflecting the benchmark-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.

Table 4: Multi-round curriculum SFT ablation. Mixture-SFT and Stage-SFT correspond to -Multi variants in Table 1; Stage-SFT uses all three stages. Bold/underline indicate best/second-best scores.

Method	BrowseComp					AMO_Bench					SealQA_0				
	Bo1	Bo2	Bo3	Bo4	Bo5	Bo1	Bo2	Bo3	Bo4	Bo5	Bo1	Bo2	Bo3	Bo4	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
Mixture-SFT	33.6	42.0	48.2	50.3	53.1	38.0	54.0	64.0	68.0	72.0	30.6	36.9	46.8	**53.6**	**54.1**
Only-Stage3	35.8	44.1	50.3	54.5	56.9	46.0	60.0	66.0	70.0	74.0	33.3	40.5	45.0	48.6	51.4
SFT-Stage1	35.7	43.7	50.9	53.2	56.4	48.0	60.0	66.0	70.0	74.0	31.5	38.7	42.3	45.0	48.6
SFT-Stage1+2	37.0	45.5	52.3	**56.4**	**59.0**	54.0	64.0	70.0	74.0	78.0	35.1	42.3	48.6	51.4	**54.1**
Stage-SFT	**38.3**	**48.4**	**53.6**	55.9	58.3	**60.0**	**68.0**	**74.0**	**78.0**	**80.0**	**38.7**	**46.0**	**52.3**	52.3	53.6

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 with checkable outcomes (Shao et al., 2024; Guo et al., 2025). However, terminal rewards provide sparse supervision for long reasoning trajectories, making credit assignment 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., 2025), outcome-grounded advantage reshaping (Li et al., 2026), and entropy-based analyses of critical decision tokens (Wang et al., 2025). 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). These methods provide denser learning signals than outcome-only supervision and help identify locally unreliable reasoning steps. Nevertheless, PRM-based approaches typically cast process information as evaluative scores or correctness labels, which may discard diagnostic details about why a trajectory deviates, what information should be reconsidered, and how subsequent reasoning should be redirected. As a result, such supervision is less directly suited to fine-grained, localized intervention and trajectory repair.

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 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 often intervene only after a full output or trial is completed. As a result, deviations can propagate before correction, introducing unnecessary inference overhead.

5 Conclusion

We proposed RLOR, an observation-guided trajectory repair framework for long-horizon reasoning. RLOR localizes the earliest deviation as a *pivot*, generates observation-based natural-language feedback, and repairs the subsequent continuation while preserving the valid prefix. Through multi-round repair supervision, curriculum SFT, and pivot-aware RL with suffix-localized credit assignment, the policy internalizes this behavior to address successive deviations without external teacher feedback at deployment. Experiments demonstrate improvements in accuracy and token efficiency. Ablations support the contributions of pivot localization, observation content, and the training design. Together, these findings highlight learning where and how to correct deviations while retaining useful progress as an effective approach to more reliable and efficient long-horizon reasoning.

REFERENCES

Shengnan An, Xunliang Cai, Xuezhi Cao, Xiaoyu Li, Yehao Lin, Junlin Liu, Xinxuan Lv, Dan Ma, Xuanlin Wang, Ziwen Wang, et al. Amo-bench: Large language models still struggle in high school math competitions. *arXiv preprint arXiv:2510.26768*, 2025.

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

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

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, et al. Critic: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Refining credit assignment in rl training of llms. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, 2025.

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

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

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024a. doi: 10.1162/tacl_a_00638.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024b.

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

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

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: a benchmark for general AI assistants. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

MiroMind Team, Song Bai, Lidong Bing, Lei Lei, Ruilin Li, Xin Li, Xiang Lin, Erquan Min, Liangcai Su, Bin Wang, Letian Wang, Shizun Wang, Xiaochen Wang, Yifan Zhang, Zhen Zhang, et al. MiroThinker-1.7 & H1: Towards heavy-duty research agents via verification. *arXiv preprint arXiv:2603.15726*, 2026.

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

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

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

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

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

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

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

Thinh Pham, Nguyen Nguyen, Pratibha Zunjare, Weiyuan Chen, Yu-Min Tseng, and Tu Vu. SealQA: Raising the bar for reasoning in search-augmented language models. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

Long Phan, Alice Gatti, Nathaniel Li, Adam Khoja, Ryan Kim, Richard Ren, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, and Dan Hendrycks. A benchmark of expert-level academic questions to assess AI capabilities. *Nature*, 649(8099):1139–1146, 2026. doi: 10.1038/s41586-025-09962-4.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

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

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

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

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

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

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

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 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)*, pp. 10014–10037, 2023.

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

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 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)*, pp. 9426–9439, 2024.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, et al. 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, 2025.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

Jian Xie, Tianhe Lin, Zilu Wang, Yuting Ning, Yuekun Yao, Tianci Xue, Zhehao Zhang, Zhongyang Li, Kai Zhang, Yufan Wu, Shijie Chen, Boyu Gou, Mingzhe Han, Yifei Wang, Vint Lee, Xinpeng Wei, Xiangjun Wang, Yu Su, and Huan Sun. QUEST: Training frontier deep research agents with fully synthetic tasks. *arXiv preprint arXiv:2605.24218*, 2026.

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

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. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025. doi: 10.48550/arXiv.2507.18071.

A Implementation Details

The pre-trained policy is available at Qwen3.5-35B-A3B. Evaluation datasets are available at BrowseComp, AMO_Bench, SealQA_0, HLE, and GAIA.

Table 5: 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 6: 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 training 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.

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 5.

A.2 SFT Training Details

We use pivot-aware trajectory supervision for SFT; hyperparameters are listed in Table 6. Both the SFT curriculum data and the RL training queries are disjoint from the evaluation test sets, preventing test-set leakage during either training stage.

A.3 Credit-Assignment Examples

Figure 5 illustrates the three trajectory cases described in Section 2.4. It distinguishes tokens receiving the main policy-gradient update from the prefix subject to weak KL regularization.

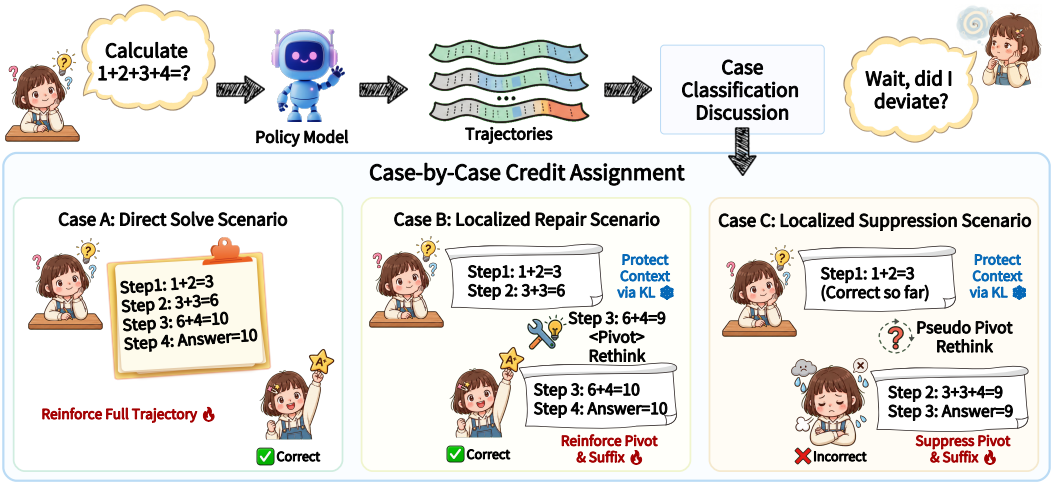


Figure 5: Case-by-case credit assignment in suffix-localized RL. RLOR classifies generated trajectories into direct-solve, localized-repair, and localized-suppression cases according to final correctness and pivot behavior. Correct no-pivot trajectories reinforce full direct solving; correct trajectories with pivot reinforce only the pivot and repaired suffix; incorrect trajectories suppress the pivot or pseudo-pivot and suffix while protecting the valid prefix.

B Additional Results

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

We provide additional Best-of-N results in Figure 6 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 6, 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.

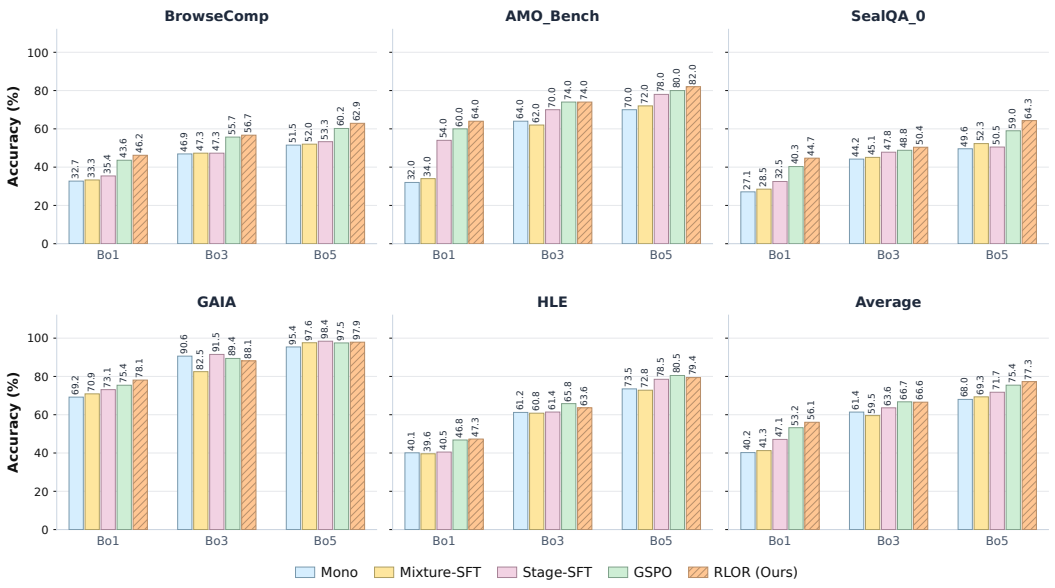


Figure 6: Best-of-N accuracy under the same-backbone setting. All methods use Qwen3.5-35B-A3B as the base model and are evaluated with the same inference-time tool interface. Results are reported for Bo1, Bo3, and Bo5 on BrowseComp, AMO_Bench, SealQA_0, GAIA, and HLE.

B.2 Additional Inference-Efficiency Comparisons

Table 7 complements the accuracy results with an inference-cost comparison of Initial ReAct (Mono), Reflexion, CRITIC, Self-Refine, and RLOR. These methods use Qwen3.5-35B-A3B and our evaluation harness.

Table 7: Additional inference-efficiency comparison under the same backbone and evaluation harness. Token and latency ratios are normalized to Initial ReAct (Mono); lower is better. Refinement-workflow latencies use a stage-wise p50 approximation.

Method	Avg. relative Qwen tokens	Approx. end-to-end p50 (s)	Relative latency
Initial ReAct (Mono)	1.00×	76.5	1.00×
Reflexion	3.07×	176.7	2.31×
CRITIC	1.99×	232.1	3.03×
Self-Refine	1.15×	257.0	3.36×
RLOR (Ours)	**0.72×**	**57.5**	**0.75×**

Measurement scope. Token counts cover Qwen inference only, excluding GPT-5.2 judge tokens and search or webpage-tool backend computation. RLOR's $0.72\times$ token ratio is obtained from the average counts in Table 2: $41{,}730.06/58{,}216.31 \approx 0.72$. For each refinement workflow, the approximate end-to-end p50 is the sum of the initial-stage p50 and the correction-stage p50, rather than the empirical median of per-query total runtimes. Relative latency is computed from the displayed times using the 76.5 s Initial ReAct baseline. RLOR's estimated latency is 57.5 s, corresponding to a relative latency of $57.5/76.5 \approx 0.75$.

Refinement overhead. In our implementation, *training-free* denotes the absence of parameter updates, while inference still involves additional model calls. After an incorrect initial answer, each Reflexion round generates a reflection, reruns the full tool-using ReAct trajectory, and invokes the judge. CRITIC runs a search-enabled verifier before revision and judging, including verification when the initial answer is correct. Self-Refine performs feedback, refinement, and judging sequentially

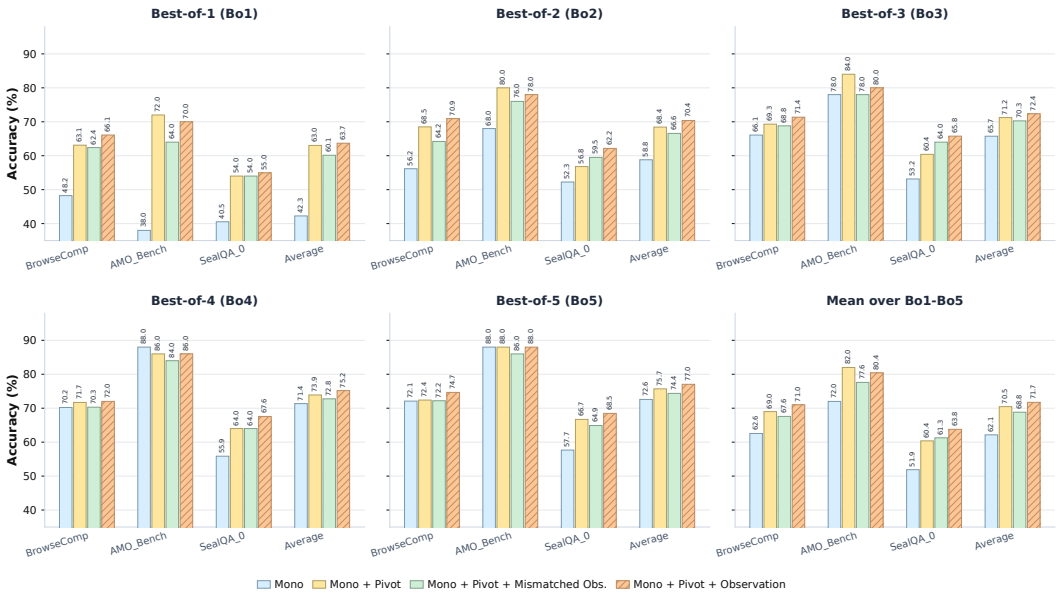


Figure 7: Ablation analysis of pivot and observation interventions on DeepSeek-V3.2-Thinking. Pivot-only intervention improves on Mono in most benchmark–budget settings. Matched observations generally outperform mismatched observations, while the relative performance of the interventions varies across benchmarks and sampling budgets.

without additional search calls. These workflows use 1.15–$3.07\times$ the Qwen tokens of Initial ReAct, with approximate latency ratios of 2.31–$3.36\times$. The different rankings by tokens and latency show that Qwen token usage alone does not capture the cost of serial calls, judging, and tool interactions.

B.3 Extended Best-of N Results on Pivot Localization and Observation Quality Ablation

To complement Figure 2, 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 2, 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.

Figure 7 shows that pivot-only intervention improves on Mono in most benchmark–budget settings. Matched observations generally outperform mismatched observations and yield the highest macro-average accuracy over the three benchmarks. The gains are not uniform: on AMO_Bench, pivot-only intervention exceeds the matched setting at Bo1–Bo3, while both fall below Mono at Bo4. These results suggest that pivot localization and aligned observations can aid trajectory repair, with benefits depending on the benchmark and sampling budget.

Inference efficiency. Figure 8 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.

B.4 Single-Round Curriculum SFT Ablation

Table 8 reports the single-round counterpart of the multi-round ablation in Table 4. It ablates Stage-SFT, which sequentially learns pivot localization, pivot-reason generation, and repaired continuation.

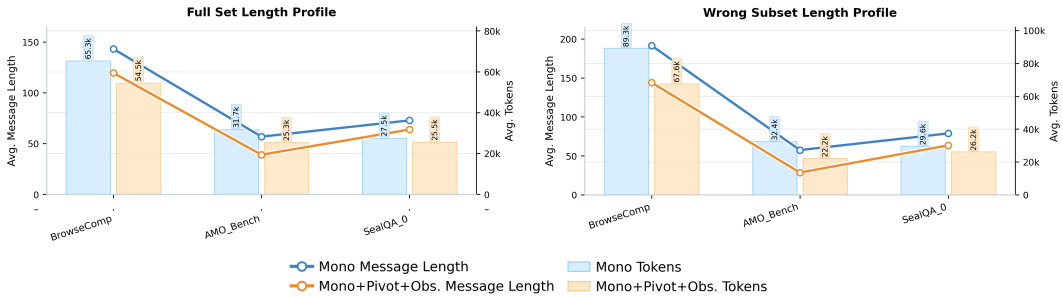


Figure 8: Efficiency analysis of pivot and observation interventions on DeepSeek-V3.2-Thinking. The left panel shows results for the full evaluation sets, and the right shows results for the subset of questions that the original model initially answered incorrectly. Bars indicate average token usage, and lines indicate the average number of messages per trajectory.

Table 8: Single-round curriculum SFT ablation. Mixture-SFT and Stage-SFT correspond to the single-round baselines in Table 1. SFT-Stage1 and SFT-Stage2 are sequential checkpoints; Stage-SFT completes all three stages, while Only-Stage3 trains only on the third-stage task. Average is the macro-average over three benchmarks. Bold and underlined scores mark the best and second-best values, respectively; tied best scores use both styles. Green shading highlights the full curriculum.

Method	BrowseComp					AMO_Bench				
	Bo1	Bo2	Bo3	Bo4	Bo5	Bo1	Bo2	Bo3	Bo4	Bo5
Base Model	32.7	42.2	46.9	48.1	51.5	32.0	56.0	64.0	68.0	70.0
Mixture-SFT	33.3	41.8	47.3	47.9	52.0	34.0	50.0	62.0	66.0	72.0
Only-Stage3	34.3	42.5	50.4	**52.6**	55.9	42.0	60.0	60.0	62.0	64.0
SFT-Stage1	34.1	42.1	50.7	52.1	**56.1**	44.0	60.0	62.0	64.0	72.0
SFT-Stage2	34.6	42.7	**50.9**	52.4	54.1	48.0	54.0	64.0	70.0	72.0
Stage-SFT	**35.4**	**44.7**	47.3	52.0	53.3	**54.0**	**64.0**	**70.0**	**72.0**	**78.0**

Method	SealQA_0					Average				
	Bo1	Bo2	Bo3	Bo4	Bo5	Bo1	Bo2	Bo3	Bo4	Bo5
Base Model	27.1	34.3	44.2	46.9	49.6	30.6	44.2	51.7	54.3	57.0
Mixture-SFT	28.5	33.4	45.1	**50.5**	**52.3**	31.9	41.7	51.5	54.8	58.8
Only-Stage3	30.6	38.5	39.1	44.7	46.6	35.6	47.0	49.8	53.1	55.5
SFT-Stage1	29.8	36.0	36.7	37.8	42.3	36.0	46.0	49.8	51.3	56.8
SFT-Stage2	**33.2**	38.8	41.5	43.3	45.1	38.6	45.2	52.1	55.2	57.1
Stage-SFT	32.5	**41.5**	**47.8**	47.8	50.5	**40.6**	**50.1**	**55.0**	**57.3**	**60.6**

The full curriculum achieves the highest macro-average Bo1 accuracy across the three benchmarks (40.6%), exceeding Mixture-SFT by 8.7 percentage points and Only-Stage3 by 5.0 points. Its macro-average accuracy is also highest at the reported Bo3 and Bo5 budgets (55.0% and 60.6%). These results support progressively organizing the repair supervision rather than mixing all curriculum tasks or training only on repaired continuation.

The gains are not uniform across individual benchmarks: on SealQA_0 at Bo1, SFT-Stage2 scores 33.2%, compared with 32.5% for the full curriculum. The progression from Stage1 through Stage2 to Stage-SFT improves macro-average Bo1 accuracy from 36.0% to 38.6% to 40.6%, but does not imply monotonic improvement in every benchmark–budget setting.

B.5 Teacher Sensitivity in Multi-Round Offline Pivot Repair

We vary the pivot annotator among GPT-5.2, Gemini-3.6-Flash, and GLM-5.2 while keeping DeepSeek-V3.2-Thinking as the student for both initial rollout and repaired continuation. In the first

repair round, the teacher localizes a pivot in the student trajectory and provides a pivot reason without access to the gold answer. Then the student then resumes from the intervention. Subsequent rounds revisit only questions that remain incorrect, identify a new pivot after the preceding pivot, and again use the student to generate the continuation. We report up to three repair rounds on BrowseComp, AMO_Bench, and SealQA_0. Repair-round count and Best-of-N sampling budget are separate axes of the evaluation.

Table 9: Teacher sensitivity in multi-round offline pivot repair with DeepSeek-V3.2-Thinking as the student. Mono is the unmodified baseline (round 0). Each repair round combines teacher pivot annotation with student continuation. Accuracy is reported in percent for Bo1–Bo5; Average is the macro-average over the three benchmarks. Bold and underlined scores mark the best and second-best teachers within each round and column, respectively; tied best scores use both styles.

Teacher	**BrowseComp**					**AMO_Bench**				
	Bo1	Bo2	Bo3	Bo4	Bo5	Bo1	Bo2	Bo3	Bo4	Bo5
Mono (round 0)	48.24	56.16	66.10	70.21	72.09	38.00	68.00	78.00	88.00	88.00
Repair round 1										
GPT-5.2	**66.10**	**70.93**	71.35	**72.01**	**74.66**	**70.00**	**78.00**	**80.00**	**86.00**	**88.00**
Gemini-3.6-Flash	65.57	70.04	**71.52**	71.90	74.19	68.00	**78.00**	**80.00**	**86.00**	**88.00**
GLM-5.2	65.35	70.23	71.45	71.99	74.42	**70.00**	**78.00**	**80.00**	84.00	86.00
Repair round 2										
GPT-5.2	**84.76**	**87.43**	**88.08**	**88.54**	**88.92**	**86.00**	**88.00**	**88.00**	**90.00**	**92.00**
Gemini-3.6-Flash	82.03	85.54	86.67	87.25	87.51	82.00	86.00	**88.00**	**90.00**	**92.00**
GLM-5.2	83.72	86.40	87.13	87.78	88.52	82.00	**88.00**	**88.00**	88.00	**92.00**
Repair round 3										
GPT-5.2	**88.15**	**88.37**	**88.81**	**89.35**	**90.24**	**94.00**	**94.00**	**96.00**	**96.00**	**98.00**
Gemini-3.6-Flash	85.67	86.08	87.19	88.63	89.85	88.00	92.00	94.00	94.00	96.00
GLM-5.2	86.59	87.13	88.42	88.45	89.74	92.00	**94.00**	**96.00**	**96.00**	96.00

Teacher	**SealQA_0**					**Average**				
	Bo1	Bo2	Bo3	Bo4	Bo5	Bo1	Bo2	Bo3	Bo4	Bo5
Mono (round 0)	40.54	52.25	53.15	55.86	57.66	42.26	58.80	65.75	71.36	72.58
Repair round 1										
GPT-5.2	**54.95**	**62.16**	**65.77**	**67.57**	**68.47**	**63.68**	**70.36**	**72.37**	**75.19**	**77.04**
Gemini-3.6-Flash	54.05	60.36	63.96	66.67	67.57	62.54	69.47	71.83	74.86	76.59
GLM-5.2	54.05	61.26	65.86	**67.57**	**68.47**	63.13	69.83	72.44	74.52	76.30
Repair round 2										
GPT-5.2	**70.27**	**72.07**	**73.56**	**73.91**	**75.06**	**80.34**	**82.50**	**83.21**	**84.15**	**85.33**
Gemini-3.6-Flash	68.47	70.27	72.07	72.97	73.91	77.50	80.60	82.25	83.41	84.47
GLM-5.2	69.37	71.17	72.97	73.56	74.77	78.36	81.86	82.70	83.11	85.10
Repair round 3										
GPT-5.2	**74.77**	**75.33**	**75.85**	**76.20**	**76.71**	**85.64**	**85.90**	**86.89**	**87.18**	**88.32**
Gemini-3.6-Flash	72.97	73.56	74.77	75.06	75.33	82.21	83.88	85.32	85.90	87.06
GLM-5.2	73.56	74.77	75.06	75.33	76.20	84.05	85.30	86.49	86.59	87.31

Table 9 shows that macro-average accuracy increases with successive repair rounds for all three teachers at every reported Best-of-N budget. With GPT-5.2 annotations, macro-average Bo1 accuracy rises from 42.26% for Mono to 63.68%, 80.34%, and 85.64% after rounds 1, 2, and 3, respectively. Gemini-3.6-Flash and GLM-5.2 show the same improvement trend, reaching 82.21% and 84.05% at round 3.

Teacher choice still affects absolute performance. At round 3, macro-average Bo1 accuracy ranges from 82.21% to 85.64% across teachers, a spread of 3.43 percentage points; the corresponding Bo5 range is 87.06%–88.32%, a spread of 1.26 points. Thus, the benefit of iterative offline repair persists

across annotators, with a smaller observed teacher gap at Bo5 than at Bo1. This comparison concerns teacher-assisted offline repair accuracy; it does not establish identical pivot annotations or unchanged performance after retraining RLOR with a different teacher.

B.6 Reliability of Pivot Annotations

We assess pivot annotation reliability through three comparisons: three independent annotation runs by GPT-5.2; agreement across GPT-5.2, DeepSeek-v4-Flash, Gemini-3.6-Flash, and GLM-5.2; and agreement between GPT-5.2 and human annotations. These comparisons evaluate annotation consistency, complementing the teacher-assisted repair accuracies in Appendix B.5. Independent annotation runs are distinct from the successive repair rounds evaluated there.

Table 10 expands the summary in Table 3 with interval overlap and chance-corrected agreement metrics.

Table 10: Agreement of pivot annotations. Agreement rates are percentages; mean distance is measured in assistant steps, and κ is unitless. Higher is better except for mean distance. Dashes indicate unreported κ values for the human comparison.

	Pivot location				
Comparison	**Exact position**	**Interval overlap**	**±1 step**	**Mean distance**	**Position κ**
GPT-5.2 repeats	88.9	88.9	92.2	2.711	0.868
Cross-model	74.4	74.4	80.9	4.524	0.702
GPT-5.2 vs. human	76.7	76.7	90.4	3.833	–

	Error category and reason		
Comparison	**Category agreement**	**Category κ**	**Reason semantic agreement**
GPT-5.2 repeats	88.9	0.816	85.6
Cross-model	81.6	0.783	73.7
GPT-5.2 vs. human	90.2	–	73.3

Metrics. Exact position agreement requires identical pivot step indices. Interval overlap counts two candidate pivot ranges as agreeing when they share at least one assistant step. The ±1-step metric allows the two pivot starts to differ by at most one assistant step, while mean distance measures the average absolute difference between their step indices. Position Fleiss' κ measures chance-corrected agreement on pivot locations.

Category agreement requires identical deviation-category labels, and category Fleiss' κ measures the corresponding chance-corrected agreement. Reason semantic agreement records whether the judging models agree that two pivot reasons identify the same underlying error, even when their wording or assigned categories differ. Table 10 reports all eight metrics.

Repeated GPT-5.2 annotation achieves 88.9% exact-position agreement and 92.2% agreement within one step. Across the four models, these values are 74.4% and 80.9%, respectively, with 81.6% category agreement. The human comparison yields 76.7% exact-position agreement, 90.4% agreement within one step, and 90.2% category agreement. Thus, allowing a neighboring step increases agreement, while precise pivot boundaries remain a source of disagreement. Reason semantic agreement is 85.6% for repeated GPT-5.2 annotation, 73.7% across models, and 73.3% against human annotations. The results support annotation repeatability and quantify the remaining differences across annotators; they do not imply identical pivot labels or reasons.

B.7 Effect of Trajectory Source on Curriculum SFT

We compare curriculum SFT using trajectories generated by DeepSeek-V3.2-Thinking or Qwen3.5-35B-A3B for two policy models, Qwen3.5-35B-A3B and Qwen3.6-35B-A3B. The policy model is

the model being fine-tuned and evaluated, whereas the trajectory model provides the source rollouts for SFT supervision. This distinguishes the trajectory source from the pivot-annotation teacher. Table 11 compares Mono, Mixture-SFT, and Stage-SFT across five benchmarks.

Table 11: Effect of the training-trajectory source on curriculum SFT. Each block fixes the policy model being fine-tuned and evaluated. Mono is the base-policy baseline before SFT and has no training-trajectory source. Scores are accuracies in percent; Average is the macro-average over five benchmarks. Bold and underlined scores mark the best and second-best values within each policy block and column, respectively; tied best scores use both styles.

Trajectory source	Method	Browse Comp	AMO_ Bench	SealQA_ 0	HLE	GAIA	Avg.
Policy: Qwen3.5-35B-A3B							
–	Mono	32.7	32.0	27.1	40.1	69.2	40.2
DeepSeek-V3.2-Thinking	Mixture-SFT	33.3	34.0	28.5	39.6	70.9	41.3
	Stage-SFT	<u>35.4</u>	<u>54.0</u>	<u>32.5</u>	<u>40.5</u>	73.1	<u>47.1</u>
Qwen3.5-35B-A3B	Mixture-SFT	34.5	38.0	30.6	40.2	<u>73.3</u>	43.3
	Stage-SFT	**37.4**	**58.0**	**35.1**	**42.7**	**76.4**	**49.9**
Policy: Qwen3.6-35B-A3B							
–	Mono	39.6	46.0	27.0	31.1	<u>84.9</u>	45.7
DeepSeek-V3.2-Thinking	Mixture-SFT	<u>41.7</u>	<u>48.0</u>	<u>28.8</u>	<u>31.4</u>	84.3	<u>46.8</u>
	Stage-SFT	**44.3**	**58.0**	**29.7**	**31.8**	**86.6**	**50.1**
Qwen3.5-35B-A3B	Mixture-SFT	35.2	40.0	24.3	25.8	80.3	41.1
	Stage-SFT	37.6	<u>48.0</u>	27.9	29.5	80.1	44.6

For the Qwen3.5-35B-A3B policy, switching from DeepSeek-V3.2-Thinking trajectories to Qwen3.5-35B-A3B trajectories increases the reported macro-average from 41.3% to 43.3% for Mixture-SFT and from 47.1% to 49.9% for Stage-SFT. The corresponding gains are 2.0 and 2.8 percentage points. For the Qwen3.6-35B-A3B policy, the same source change instead reduces the reported macro-average from 46.8% to 41.1% for Mixture-SFT and from 50.1% to 44.6% for Stage-SFT, a decrease of 5.7 and 5.5 points in each case.

Stage-SFT has a higher macro-average than Mixture-SFT in all four policy–source combinations. However, the preferred trajectory source differs between the two policy models. These results indicate that trajectory-source choice matters for the evaluated policies, while the macro-average benefit of staged supervision persists across both sources. They do not support a claim that replacing the trajectory generator leaves performance unchanged.

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:
```

Prompt for Expert-Label Annotation, continued

```
    - 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.
```

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 student 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.

C.1 A Rollout Example with Pivot and Observation Interventions

Figure 9 presents an example rollout from the DeepSeek-V3.2-Thinking experiments under pivot and observation interventions (Appendix B.3). The pivot localizes unsupported anchoring on an association, and the accompanying observation guides subsequent evidence gathering. Selected excerpts show the retained context, intervention feedback, subsequent searches and tool evidence, and final answer.

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.

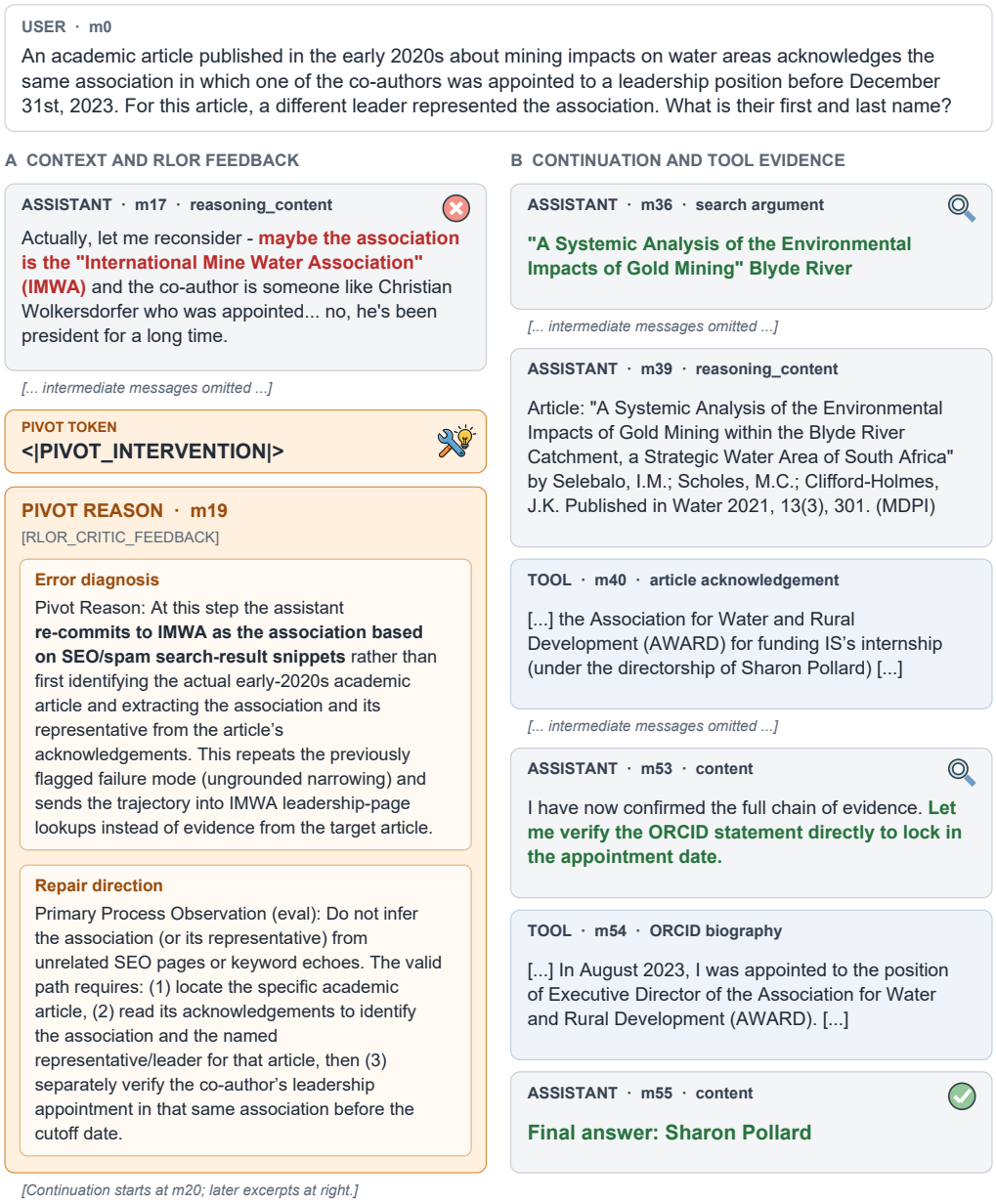


Figure 9: An example rollout of DeepSeek-V3.2-Thinking under pivot and observation interventions. The excerpts show the pivot, error diagnosis, repair guidance, and subsequent continuation. Red and green bold text mark the unsupported hypothesis and repaired actions/final answer, respectively; message indices are zero-based.

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 constructs an unrelated Bundesliga hypothesis 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 final answer is:

```
109th minute, UEFA Euro 2016.
```

This matches the gold answer.

Table 12 summarizes the key differences between the original and repaired trajectories.

Table 12: 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.

Aspect	**Original trajectory**	**Pivot-repaired trajectory**
Initial reasoning	Extracts constraints on 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 search evidence.	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.

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 standardized 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 131072 \
    --tensor-parallel-size 8 \
    --served-model-name expert \
    --trust-remote-code \
    --disable-log-requests
```

F THE USE OF LARGE LANGUAGE MODELS (LLMS)

We used LLMs as writing aids to polish language (e.g., grammar, wording, and clarity) for text written by the authors. We also used LLMs to assist in constructing synthetic training data, including trajectory annotation, pivot localization and reasoning, and the generation and evaluation of repaired continuations, as described in the methodology and appendix. All suggestions from LLMs were reviewed by the authors and accepted, modified, or discarded at our discretion. No passages were inserted without author verification, and all references and factual claims were curated and checked by the authors.

The authors take full responsibility for the contents of this paper, including any text refined with LLM assistance and synthetic data generated with LLMs. LLMs are not eligible for authorship and are not listed as authors. No non-public or sensitive data were provided to the LLM service beyond manuscript text and data used for the research described in this paper, and our usage adhered to relevant institutional and venue policies on LLM use.