Learning from Peers in Reasoning Models

Anonymous ACL submission

Abstract

While Large Reasoning Models (LRMs) possess intrinsic self-correction capabilities, our study **quantifies** a critical failure mode—the *"Prefix Dominance Trap"*—where flawed initial steps irreversibly derail recovery **regardless of error explicitness**. To transcend isolated self-correction, we draw inspiration from collaborative psychology to propose **Learning from Peers (LeaP)**, an **in-process** framework where parallel paths periodically exchange compressed insights to correct errors through diversity, supported by **actionable design guidelines** (granularity, routing, timing). We further introduce and **release open-source LeaP-T models (1.5B–14B)** to empower collaborative reasoning in smaller models. Systematic analysis of communication dynamics establishes principled guidelines ensuring robust peer collaboration. Extensive evaluations on AIME 2024/2025, AIMO, and GPQA Diamond demonstrate significant robustness gains: QwQ-32B with LeaP outperforms its baseline by nearly 5 points and surpasses DeepSeek-R1-671B, while LeaP-T-7B matches the 14B baseline. Code will be open-sourced.

1 Introduction

Large reasoning models (LRMs) (OpenAI, 2024b,c, 2025b; Guo et al., 2025; Team, 2025) demonstrate strong performance on complex tasks through test-time scaling (Zhang et al., 2025; Snell et al., 2024; Zeng et al., 2025b; Chen et al., 2024), leveraging emergent self-verification and self-refinement ("aha moments") (Guo et al., 2025). However, recent research reveals notable limitations in this self-correction mechanism (Zeng et al., 2025b): LRMs frequently become stuck in incorrect reasoning paths that are rarely corrected.

The Prefix Dominance Trap. To quantify this fragility, we design a diagnostic task where models must solve problems from fixed prefix drawn from both correct and incorrect responses. Surprisingly, many LRMs (including QwQ-32B) suffer a nearly 20% performance drop when starting from a flawed beginning constituting just 15% of the response length. We term this phenomenon—where *a short, flawed prelude irreversibly derails reasoning*—the *"Prefix Dominance Trap"*. This reveals that LRMs commit to early paths as "ground truth", motivating us to look beyond internal self-correction and instead leverage external signals for recovery.

Cross-Path Correction via LeaP. Inspired by collaborative psychology (Giuliodori et al., 2006; Snyder et al., 2015), we propose **Learning from Peers** (LeaP), enabling parallel reasoning paths to communicate during inference. Concretely, after every T tokens, each path summarizes its intermediate reasoning (256 tokens) and shares it with peers via a heuristic routing mechanism (Dispersed, Clustered, or Hybrid). This shifts the cognitive burden from generating correct paths to verifying and refining diverse peer trajectories. Unlike multi-agent debate methods (Du et al., 2024; Liang et al., 2024; Sun et al., 2025; Xu et al., 2023; Li et al., 2024; Wang et al., 2024) that refine full answers through post-hoc discussion, LeaP operates *within* the generation process to enable timely error correction rather than consensus-seeking (Figure 2).

Comprehensive Evaluation. We first validate our hypothesis under the *"Prefix Dominance Trap"* setting, LeaP reduces the performance gap by nearly 10%. Then we conduct evaluations **without any prefix** on standard reasoning benchmarks (AIME 2024/2025, AIMO, GPQA Diamond), LeaP significantly improves Pass@1, Cons@N and Pass@N over independent reasoning, with QwQ-32B surpassing the much larger DeepSeek-R1-671B. Results on ZebraLogic (Lin et al., 2025) confirm generalization beyond STEM.

Adaptive Training for Communication. During these experiments, we observe that smaller models without further training (e.g., 7B) struggle with zero-shot summarization (Case studies in Appendix D.2). We introduce **LeaP-T**, fine-tuned on high-quality peer-interaction trajectories distilled from larger models. LeaP-T-7B matches DeepSeek-R1-Distill-Qwen-14B on AIME 2024, demonstrating that collaborative reasoning can be induced through targeted adaptation.

Systematic Analysis. Beyond proposing the mechanism, we conduct **the first** comprehensive empirical study on *how* peer communication should be structured specifically for reasoning models—a dimension largely unexplored by prior multi-agent debate methods. Through extensive ablations on communication granularity (T), traffic (Top-k), temporal dynamics, and error tolerance, we derive actionable guidelines. These insights transform peer learning from a black-box ensemble into a principled, tunable inference paradigm with predictable scaling properties.

Contributions. **I) Diagnostic:** We quantify the Prefix Dominance Trap, showing that short, low-quality prefixes severely degrade performance regardless of error explicitness. **II) Method:** We propose LeaP, a framework for *in-process* cross-path correction with actionable design guidelines (granularity, routing, timing). **III) Artifacts:** We release open-source LeaP-T models (1.5B-14B) to facilitate research on collaborative reasoning. **IV) Guidelines for Peer Collaboration:** We provide systematic analysis of communication dynamics in reasoning models, establishing actionable guidelines for granularity, routing, timing, and robustness that ensure predictable peer collaboration.

2 Related Work

Self-Correction Bottleneck. Before the era of LRMs, self-correction relied on explicit prompting strategies: models were instructed to iteratively critique and refine outputs (Self-Refine (Madaan et al., 2023), CoSC (Gao et al., 2024)), decompose verification into sub-questions (CoVe (Dhuliawala et al., 2024)), or leverage external tools for validation (CRITIC (Gou et al., 2023)). With the advent of reasoning models, LRMs now exhibit emergent self-correction without explicit instructions (Guo et al., 2025). However, recent studies reveal the *fragility* of this intrinsic capability: performance does not consistently improve with chain length (Zeng et al., 2025b; Marjanović et al., 2025), and models frequently become entrenched in incorrect paths (Zeng et al., 2025b)—a phenomenon we term the *Prefix Dominance Trap*.

From Ensembling to Collaboration. Standard test-time scaling aggregates independent outputs via Self-Consistency (Wang et al., 2022) or Best-of-N (Cobbe et al., 2021). Recent work explores interactive collaboration: multi-agent debate (Du et al., 2024; Liang et al., 2024; Sun et al., 2025) refines final answers through iterative discussion; ReConcile (Chen et al., 2023) and Corex (Sun et al., 2023) employ multi-round deliberation with role specialization; MoA (Wang et al., 2024; Li et al., 2024) aggregates hierarchical layer-wise outputs. Unlike these post-hoc methods that operate on complete answers, LeaP enables *in-process* communication via intermediate summaries.

3 Enabling Cross-path Correction in Parallel Inference: LeaP

3.1 Motivation: Prefix Dominance Trap

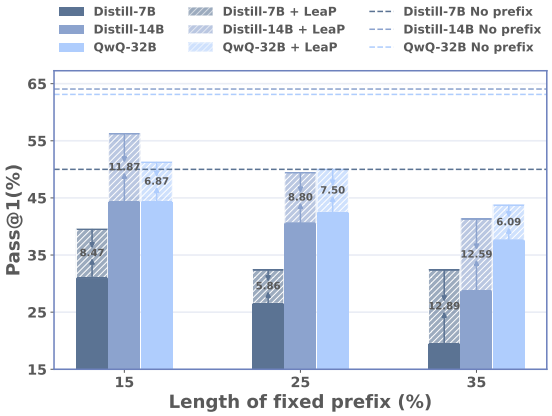


Figure 1: The results of starting with and without bad beginnings.

To quantify self-correction fragility, we design a prefix-constrained generation task. If an LRM possesses robust correction abilities, it should recover from flawed starts and narrow the accuracy gap with unconstrained generation. We extract initial segments (15%–35% of response length) from incorrect trajectories generated by DeepSeek-R1-Distill-Qwen and QwQ-32B on AIME 2024 (MAA, 2024), then force models to continue from these prefixes at $\tau = 1$. We further use gpt-oss-120b (OpenAI, 2025a) to taxonomize incorrect prefixes by whether they contain *explicit* erroneous conclusions or merely *implicit* flawed reasoning. We find that 55 to 65% of prefixes contain only

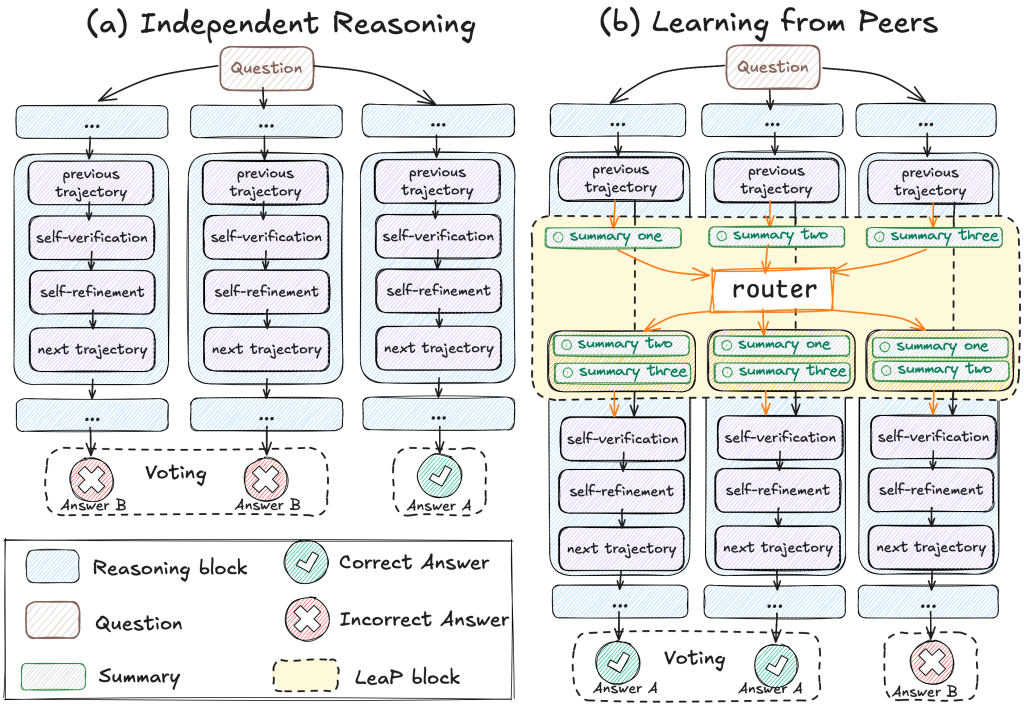


Figure 2: The illustration of (a) Independent Reasoning and (b) the proposed method Learning from Peers (LeaP). In independent reasoning, multiple paths are generated independently in parallel. In contrast, LeaP inserts a LeaP block into reasoning path, encouraging the model to learn from peers.

implicit flaws without explicit errors. So self-correction capability is measured by the Pass@1 accuracy gap between prefix-constrained and unconstrained generation.

The Trap Emerges. Despite these short prefixes, all LRMs exhibit a devastating 20%–30% performance drop (Figure 1). This reveals **limited self-correction**: once committed to an initial flawed trajectory, models rarely recover. We term this the *"Prefix Dominance Trap"*—where a brief, flawed prelude irreversibly derails reasoning.

Findings 1

LRMs exhibit fragile self-correction and fail to recover from context with explicit errors or implicit biases.

3.2 Methodology: Cross-path Correction

LeaP Block Overview. To mitigate the Prefix Dominance Trap, we introduce **Learning from Peers** (LeaP), an inference-time strategy where parallel reasoning paths periodically exchange compressed insights. As shown in Figure 2b, LeaP inserts *LeaP blocks* every T tokens into the inference process. Each block executes: (1) **Summarization**: condensing current reasoning into shareable messages; and (2) **Routing**: selecting top-k relevant peer summaries to inject into subsequent generation. Here, T controls communication frequency, while k regulates information traffic.

Stage I: Summarization. Each path compresses its intermediate approach, key insights, and partial results into a concise summary (strictly $\leq$256 tokens). A randomly selected prompt template (comprising a summary trigger and template, see Appendix A) directs this condensation, ensuring variability while maintaining core information capture. This compression transforms intractable full-context sharing into efficient peer communication.

Stage II: Routing. Direct broadcast of all summaries to all paths incurs prohibitive overhead. Instead, a routing function $\mathcal{R}$ selects for each path i a top-k subset $\mathcal{C}_i$ from peer summaries $\{s_j\}_{j \neq i}$. We explore three heuristics:

$$\mathcal{C}_i = \begin{cases} \text{Top-}k\{\,1 - \text{sim}(s_i, s_j)\,\}, & \text{Dispersed} \\ \text{Top-}k\{\,\text{sim}(s_i, s_j)\,\}, & \text{Clustered} \\ \text{Top-}\frac{k}{2}\{\,\text{sim}\,\} \cup \text{Top-}\frac{k}{2}\{\,1 - \text{sim}\,\}, & \text{Hybrid} \end{cases} \tag{1}$$

where $\text{sim}(s_i, s_j) = 1 - \frac{D_{\text{lev}}(s_i, s_j)}{\max(|s_i|, |s_j|)}$ denotes normalized Levenshtein similarity. Intuitionally, Dispersed routing prioritizes dissimilar perspectives to break erroneous patterns; Clustered reinforces aligned trajectories; Hybrid balances exploration and exploitation.

Illustrative Mechanism. Figure 3 demonstrates peer-triggered correction: Path i (incorrect, red) receives a summary from Path j (correct, green) via the routing mechanism. By incorporating this external insight at the LeaP block, Path i identifies

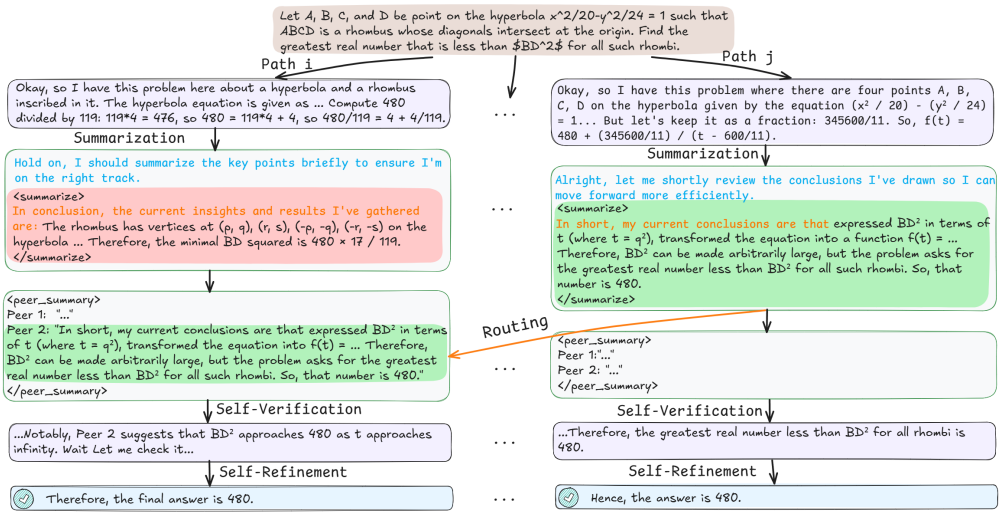


Figure 3: An example of how LeaP enables communication between path i and j. Text in red indicates the current path is incorrect. Text in green shows a correct summary received from a peer.

its error and transitions toward the correct solution, exemplifying how timely peer intervention enables error recovery during generation.

3.3 Can LeaP help in "Prefix Dominance Trap"?

Applying LeaP to the Prefix Dominance Trap setting (Section 3.1), we observe consistent reduction in the performance gap across all model sizes (Figure 1). For instance, on DeepSeek-Distill-Qwen-14B, LeaP narrows the accuracy drop from 19.88 to 7.81 percentage points. Case studies in Appendix D.1 compare LeaP with the baseline. Remarkably, LeaP also outperforms independent reasoning when starting from *good* prefixes likely to lead correct answers (Appendix B). This indicates that peer interaction enhances reasoning robustness regardless of initial trajectory quality.

Findings 2

LRMs can timely correct error and refine by strong self-verification capabilities to assess peer reasoning paths.

4 Evaluating LeaP on Reasoning Benchmarks

We assess the effectiveness of LeaP across diverse reasoning tasks **without fixing prefixes**. This section details the experimental setup followed by a comprehensive analysis of the results.

4.1 Experimental Setup

We conduct experiments on AIME 2024 (MAA, 2024), AIME 2025 (MAA, 2025), and the AIMO 2025 (Frieder et al., 2024) and GPQA Diamond (Rein et al., 2024). Performance is measured using Pass@1, Pass@k (k=2,4,8) and Cons@N (voting among N generations). We deploy the DeepSeek-R1-Distill-Qwen (MIT License) series (7B, 14B) and QwQ-32B (Team, 2025) and utilize vLLM (Kwon et al., 2023) for inference. We compare LeaP against independent reasoning and two self-correction baselines (Huang et al., 2023; Gao et al., 2024). Following protocols (Team, 2025), we set the temperature to 0.6, Top-p to 0.95, and Top-k to 40. For sampling, we set $N = 32$ for math tasks and $N = 8$ for GPQA, with maximum token of 16k for 7B/14B and 32k for the 32B models.

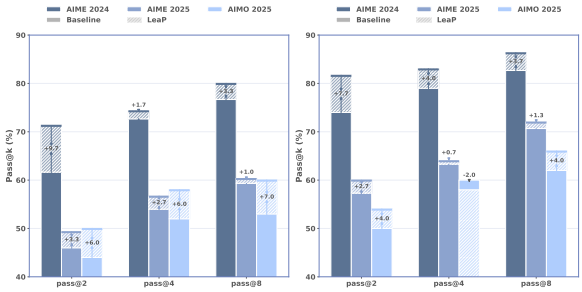
4.2 Results

Broad Performance Gains. Table 1 demonstrates that LeaP consistently improves Pass@1 across all benchmarks and model scales. Notably, QwQ-32B with Top-4 *Dispersed* routing surpasses the significantly larger DeepSeek-R1-671B on all three math tasks, validating that LeaP enhances single-path reasoning quality rather than merely exploiting ensemble effects. We further find that increasing the number of peer summaries from Top-2 to Top-4 consistently yields performance gains, suggesting that models benefit from aggregating a broader range of thinking paths. Among the routing strategies, *Dispersed* and *Hybrid* clearly outperform *Clustered*. Moreover, we demonstrate the **generalization** of LeaP to non-STEM reasoning tasks in Appendix H.2. Furthermore, we also show LeaP against specialized self-correction (Gao et al., 2024) and MoA (Wang et al., 2024) baselines in Appendix F.

Beyond Ensemble: Pass@k Validation. To ensure that the observed improvements arise from enhanced reasoning quality rather than merely increased sampling, we evaluate *Pass@k* ($k = 2, 4, 8$). For each setting, experiments are repeated

Table 1: We evaluate LeaP against the baseline on AIME 2024, AIME 2025, AIMO 2025, and GPQA Diamond, where our method shows significant outperformance. For comparison, we also report results for DeepSeek-R1-671B. As it lacks an official AIMO 2025 score, we evaluated it via four inference runs on the official website. R1-7B and R1-14B are the 7B and 14B versions of DeepSeek-R1-Distill-Qwen, respectively.

Benchmarks	Models	Independent Reasoning	Self-Correct Prompt	Clustered		Hybrid		Dispersed		R1-671B
				Top-2	Top-4	Top-2	Top-4	Top-2	Top-4	
AIME 2024	R1-7B	51.35	52.81	56.15	59.27	59.17	61.67	60.31	60.52	79.8
	R1-14B	64.47	65.63	72.08	71.04	71.77	74.48	71.15	77.29	
	QwQ-32B	79.69	78.32	78.96	81.56	81.88	81.67	81.56	85.83	
AIMO 2025	R1-7B	37.50	37.81	37.81	39.06	41.88	40.31	39.06	45.00	65.0
	R1-14B	46.87	45.54	51.50	55.31	51.88	52.19	49.38	51.25	
	QwQ-32B	63.75	64.38	63.75	65.63	64.06	64.69	67.19	67.19	
AIME 2025	R1-7B	37.81	35.73	36.98	37.50	37.19	39.27	40.93	38.44	70.0
	R1-14B	48.64	48.13	50.42	51.88	54.38	54.38	50.31	54.17	
	QwQ-32B	68.13	68.96	68.85	71.35	71.04	72.50	70.83	71.67	
GPQA Diamond	R1-7B	46.91	47.66	52.97	52.65	51.83	53.47	53.28	55.56	71.5
	R1-14B	53.47	53.09	54.80	58.33	54.42	57.89	55.68	55.05	
	QwQ-32B	58.00	58.52	65.03	65.28	61.87	65.21	66.16	63.32	
Avg.	R1-7B	43.39	43.50	45.98	47.12	47.52	48.68	48.40	49.88	71.58
	R1-14B	53.36	53.09	57.20	59.14	58.11	59.74	56.63	59.44	
	QwQ-32B	67.39	67.55	69.15	70.96	69.71	71.02	71.44	72.00	

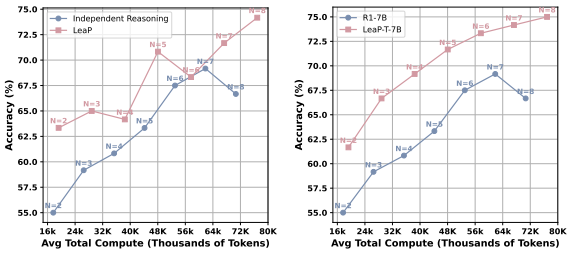


(a) 7B (b) 14B

Figure 4: Pass@N accuracy on AIME 2024.

five times and the reported results are averaged. As shown in Figure 4, LeaP consistently outperforms independent sampling baselines across nearly all k. The performance gap further widens as k increases, suggesting that peer collaboration explores the solution space more effectively than naive parallel sampling and yields more diverse correct solutions.

Efficiency Under Compute Constraints. Figure 5a plots Cons@N accuracy against total token consumption. LeaP achieves higher accuracy than independent reasoning under identical compute budgets, confirming that communication overhead is justified by superior consensus quality. Detailed cost breakdowns are provided in Appendices H.1.



(a) LeaP (b) LeaP-T

Figure 5: Cons@N accuracy vs. Total computational cost (tokens) on AIME 2024.

5 Adapting LeaP to Small Models

Training the Collaborative Paradigm. Smaller models struggle to follow summarization and reflection instructions in zero-shot settings (Appendix D.2). To bridge this gap, we introduce **LeaP-T**, which explicitly teaches collaborative reasoning via supervised fine-tuning. We construct a dataset of ∼1,000 AIME problems (1984–2023), using DeepSeek-R1-Distill-Qwen-32B as a teacher to generate LeaP trajectories (Top-4 *Dispersed*, T=4096). We filter for correct final answers and concise peer summaries (<256 tokens), then fine-tune 1.5B, 7B, and 14B models. Critically, we compare against a baseline SFT using the same teacher outputs but *without* the LeaP structure.

Interaction Over Distillation. Table 2 shows LeaP-T consistently outperforms both the base model and standard SFT across all scales. Notably, standard SFT from the 32B teacher fails to improve (or degrades) performance, as the base models are already distilled from stronger R1-671B. This confirms that LeaP-T gains from learning the **collaborative paradigm** rather than mere knowledge transfer. For instance, LeaP-T-7B matches DeepSeek-R1-Distill-Qwen-14B on AIME 2024.

Efficient Test-Time Scaling. Figure 5b demonstrates that LeaP-T-7B achieves superior Cons@N scaling compared to baselines under identical token budgets, confirming that trained peer-collaboration capabilities translate to efficient compute utilization.

6 In-depth Analysis

To gain a deeper understanding of how and why communication mechanism work in LRMs and provide a clear guideline for deploy LeaP, we con-

Table 2: Evaluation of our LeaP-T from 1.5B to 14B on three math benchmarks.

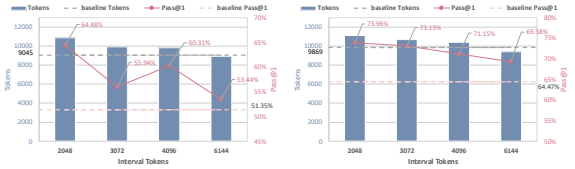
Models	AIME 2024		AIME 2025		AIMO 2025		Avg.	
	Pass@1	Cons@32	Pass@1	Cons@32	Pass@1	Cons@32	Pass@1	Cons@32
R1-1.5B	32.00	50.00	24.69	30.00	14.00	30.00	23.56	36.67
+ SFT	31.04	**56.67**	23.23	36.67	15.31	30.00	23.19	41.11
+ LeaP	34.90	46.67	26.46	30.00	15.63	30.00	25.66	35.56
LeaP-T-1.5B	**37.08**	**56.67**	**26.67**	**40.00**	**20.31**	**40.00**	**28.02**	**45.56**
R1-7B	51.35	73.33	37.81	50.00	37.50	50.00	42.22	57.78
+ SFT	51.56	**80.00**	35.73	53.33	33.75	40.00	40.35	57.78
+ LeaP	60.52	76.67	38.44	53.33	**45.00**	50.00	47.99	60.00
LeaP-T-7B	**64.38**	**80.00**	**41.25**	**56.67**	44.06	**60.00**	**49.90**	**65.56**
R1-14B	64.47	80.00	48.64	60.00	46.87	60.00	53.33	66.67
+ SFT	65.63	**83.33**	46.88	63.33	45.63	60.00	52.71	68.89
+ LeaP	**77.29**	**83.33**	54.17	60.00	51.25	60.00	60.90	67.78
LeaP-T-14B	76.46	**83.33**	**54.27**	**70.00**	**52.50**	60.00	**61.08**	**71.11**

duct a comprehensive analysis: communication sensitivity (Section 6.1), robustness (Section 6.2), computational overhead (Appendix H.1), homogenization (Appendix H.3) and the design of routers (Appendix H.4).

6.1 Sensitivity Analysis of Communication

To better deploy LeaP in practice, we investigate the sensitivity of communication on several factors, including granularity T (Section 6.1.1), traffic (Section 6.1.2), evolution tendency of types (Section 6.1.3), and position (Section 6.1.4).

6.1.1 On Communication Granularity T



(a) 7B (b) 14B

Figure 6: Pass@1 and total tokens on AIME 2024 for 7B and 14B models with LeaP, evaluated across interval tokens of LeaP from 2048 to 6144.

To study communication granularity T, we vary the T between LeaP blocks for DeepSeek-R1-Distill-Qwen 7B and 14B models, using a fixed top-2 *Dispersed* routing strategy. Figure 6 shows that on AIME 2024, less frequent communication (larger T) slightly decreases Pass@1 performance. Concurrently, larger T (less frequent communication) consumes fewer tokens, as each LeaP block incurs token costs for summarization and routing. Results on LeaP-T (Appendix H.6.2) further validate this finding.

Findings 3

Communication granularity directly tunes the accuracy-efficiency trade-off.

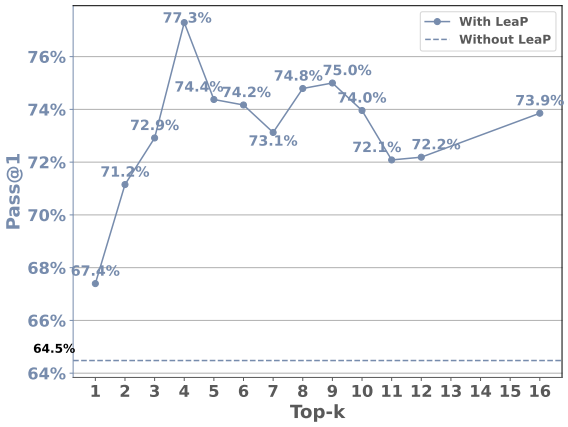


Figure 7: The top-k Performance of LeaP-14B on AIME 2024, evaluated across the top-k from 1 to 16.

6.1.2 On the Communication Traffic (Top-k)

Contrary to the intuition that more peer summaries always improve performance, we find that $k = 4$ yields the peak Pass@1 score, with performance declining for $k > 4$ (Figure 7). Insufficient information limits perspective diversity, while excessive summaries introduce noise and overwhelm reasoning capacity. This establishes $k = 4$ as the effective balance between exploration richness and cognitive coherence.

Findings 4

Moderate peer traffic optimizes error correction; excessive communication undermines reasoning quality.

6.1.3 On Evolution Tendency of Communication Types

Building on prior findings of optimal communication traffic, we analyze the evolution of communication *types* to understand peer learning dynamics. For this analysis, we fix the communication interval $T = 4$K tokens and use the top-4 *Dispersed* routing strategy. We categorize communication outcomes as: *Consistent* (paths align with peers before and after communication), *Unaffected* (paths differ from

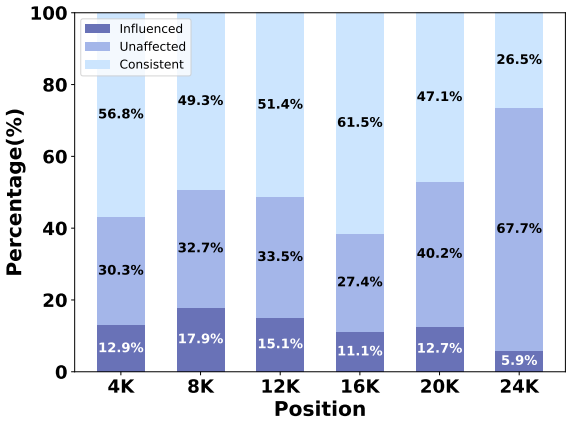


Figure 8: Communication types distribution of QwQ-32B on AIME 2024 at different positions, categorized into *Consistent*, *Unaffected*, and *Influenced*.

peers and remain unchanged), or *Influenced* (paths initially differ but adjust after peer input). GPT-4o (OpenAI, 2024a) performs annotation, validated by humans on 10% of samples (≈120) with over 95% agreement. Figure 8 plots this distribution for QwQ-32B on AIME 2024. Late in reasoning, *Unaffected* cases rise significantly.

Findings 5

Early-to-mid reasoning stages offer the optimal window for peer intervention.

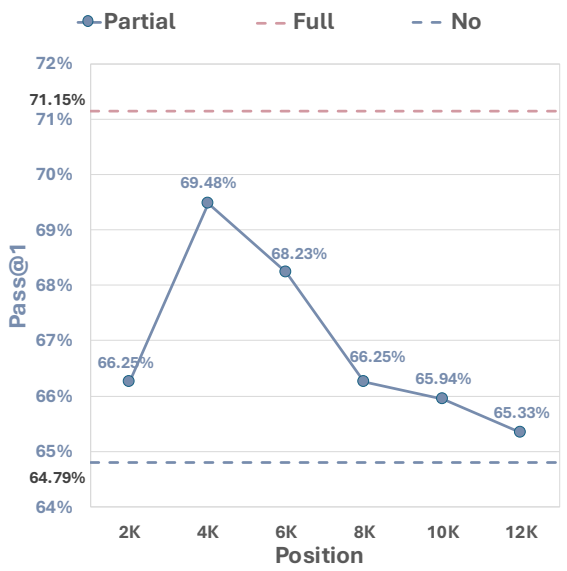


Figure 9: Performance of partial LeaP on the 14B model.

.

6.1.4 On the Communication Position (When only Communicating Once)

To further analyze sensitivity to communicating at different positions, we introduce LeaP-S where communication occurs only once during the reasoning process. In Figure 9, we report results of DeepSeek-R1-Distill-Qwen-14B (Guo et al., 2025) on AIME 2024 (MAA, 2024) under this single communication setting. We observe that performance improves when communication occurs early in the reasoning process, increasing from 66.25 at 2K tokens to a peak of 69.48 at 4K tokens. However, beyond this point, performance declines, dropping to 65.33 by the end. These results suggest that early-stage communication is more effective than late-stage interaction.

Findings 6

Single early communication suffices for significant gains; late-stage peer exposure yields diminishing returns.

Table 3: Comparison of LeaP-S with independent reasoning (`Cons@k`) baselines on AIME 2024.

Method	Score	Tokens
R1-14B + LeaP-S	**68.13**	23,252
R1-14B `Cons@2`	53.33	20,944
R1-14B `Cons@3`	66.67	29,319

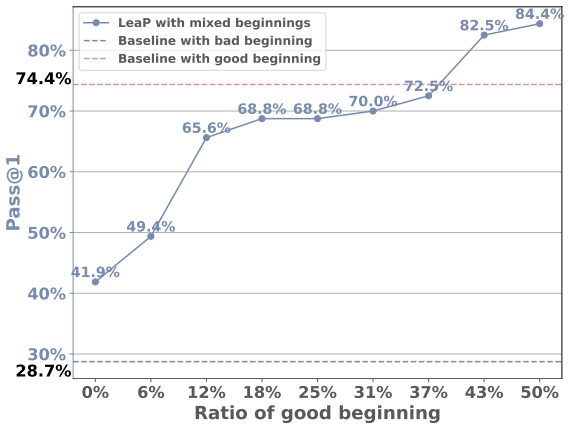
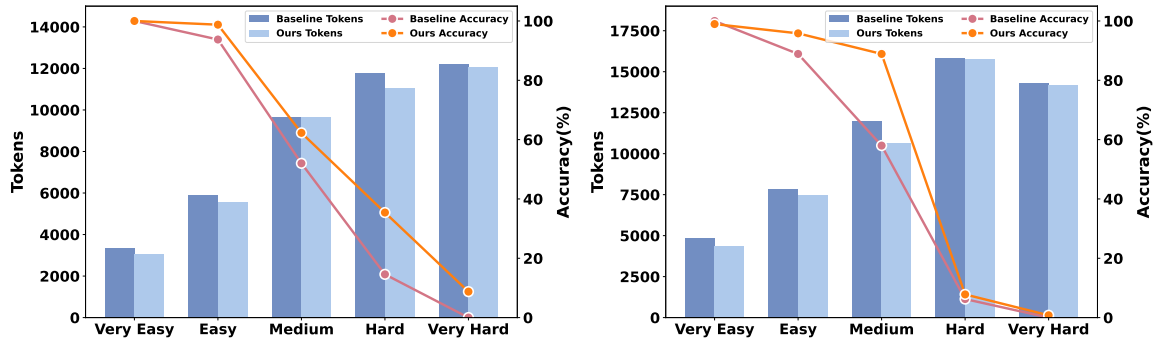


Figure 10: Pass@1 for various ratios of good beginnings, with beginning length at 30% of average response length.

Deploy LeaP Efficiently: LeaP-S. For resource-constrained scenarios, *LeaP-S* (single communication at 4K tokens) achieves 68.13% Pass@1 on AIME 2024 (14B) with only 23K tokens—surpassing both the baseline (64.47%) and Cons@3 (66.67%, 29K tokens). This demonstrates higher accuracy at lower cost than standard self-consistency (Table 3). LeaP-S yields substantial



(a) 7B (b) 14B

Figure 11: The Pass@1 and average token distribution across different difficulty levels, from Very Easy to Very Hard, for the 7B and 14B versions of DeepSeek-R1-Distill-Qwen.

improvements over the baseline. More comprehensive results is provided in Appendix G.

6.2 Post-hoc Robustness Analysis of LeaP

To better understand the reliability of LeaP in practical scenarios, we conduct a post-hoc robustness analysis. Specifically, we analyze whether the benefits of LeaP persist under two settings: (1) Vary the error path in peers (Section 6.2.1), and (2) Vary the difficulty levels (Section 6.2.2).

6.2.1 Robustness on Error Tolerance

A common concern is that low-quality paths in LeaP might mislead others, suggesting LeaP requires a high proportion of correct paths to avoid performance degradation from noise. To test LeaP's error tolerance, we vary the proportion of "good beginnings." For 10 questions, we select good and bad beginnings (initial 30% of tokens from DeepSeek-R1-Distill-Qwen-14B responses, following Section 3.1). We then construct 16 mixed initial responses per question, varying the good beginning proportion from 0% to 50%. Figure 10 reveals a surprising trend: LeaP consistently outperforms the baseline across all configurations, even with **no** good beginnings. For instance, with 0% good beginnings, LeaP's Pass@1 is 41.88, versus the baseline's 28.75. Remarkably, with only 43% good beginnings, LeaP surpasses the baseline's performance achieved when **all** its beginnings are good. These results counter the assumption that LeaP requires mostly correct paths, demonstrating its high robustness. Peers can distill useful signals from noisy summaries, attributable to the self-verification ability of LRMs.

6.2.2 Robustness at Various Difficulty Levels

Another concern is whether LeaP maintains robustness across various difficulty levels. It is possible that LeaP performs well primarily because the benchmark contains a certain proportion of simple questions. But can LeaP still provide benefits when problems become significantly more challenging?

To answer this question, we categorize AIME 2024 questions into five levels based on baseline correct responses (out of 32 from independent parallel reasoning): "Very Easy" (32 correct), "Easy" (25–31), "Medium" (9–24), "Hard" (1–8), and "Very Hard" (0). We then compute accuracy and reasoning-specific token usage (excluding peer summary tokens) per level. Figure 11 (results for DeepSeek-Distill-Qwen-7B and -14B) shows that **LeaP consistently improves accuracy across all difficulty levels**, even on "Very Hard" questions where the baseline fails completely. This suggests LeaP amplifies partial correctness and helps recover from complete failures. Furthermore, LeaP often uses fewer reasoning-specific tokens than the baseline. This, combined with LeaP's reduction of "Aha" moments (Section 4.2), indicates LeaP promotes earlier consensus and reduces overthinking. Appendix H.7 provides additional results for LeaP-T models, further confirming this robustness.

7 Conclusion

We quantify the *"Prefix Dominance Trap"*, demonstrating that LRMs suffer fragile self-correction regardless of whether initial errors are explicit or implicit. To address this, we propose **Learning from Peers** (LeaP), an *in-process* cross-path correction framework with **actionable design guidelines** (granularity, routing, timing) derived from systematic analysis of communication dynamics. We release **open-source LeaP-T models** (1.5B–14B) to democratize collaborative reasoning. Our work establishes that peer collaboration, when properly structured, transforms reasoning from isolated trial-and-error into robust, verifiable computation—opening avenues for efficient, interpretable, and scalable LRM deployment.

8 Limitations

While LeaP demonstrates strong empirical success, we identify three limitations suggesting future refinement. First, the fixed-interval communication schedule, though effective, lacks adaptability to dynamic reasoning flows; implementing uncertainty-triggered communication—where paths exchange insights only when confidence drops (e.g., high token entropy) or verification is needed—would reduce unnecessary overhead while preserving timely error correction. Second, current routing relies on heuristic similarity (Levenshtein distance), which may overlook semantic nuances; adopting embedding-based relevance metrics or lightweight learned routers could better match peers with complementary reasoning. Third, LeaP depends on high-quality explicit summaries, incurring token overhead and compression loss; a promising architectural direction is latent-space peer communication, where parallel paths exchange information via cross-path attention mechanisms, eliminating token costs and enabling end-to-end optimization of collaborative reasoning.

References

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2023. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. *arXiv preprint arXiv:2309.13007*.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. 2024. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Zhipeng Chen, Yingqian Min, Beichen Zhang, Jie Chen, Jinhao Jiang, Daixuan Cheng, Wayne Xin Zhao, Zheng Liu, Xu Miao, Yang Lu, Lei Fang, Zhongyuan Wang, and Ji-Rong Wen. 2025. An empirical study on eliciting and improving r1-like reasoning models. *arXiv preprint arXiv:2503.04548*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the association for computational linguistics: ACL 2024*, pages 3563–3578.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first international conference on machine learning*.

Simon Frieder, Sam Bealing, Arsenii Nikolaiev, Geoff C. Smith, Kevin Buzzard, Timothy Gowers, Peter J. Liu, Po-Shen Loh, Lester Mackey, Leonardo de Moura, Dan Roberts, D. Sculley, Terence Tao, David Balduzzi, Simon Coyle, Alex Gerko, Ryan Holbrook, Addison Howard, and XTX Markets. 2024. Ai mathematical olympiad - progress prize 2. https://kaggle.com/competitions/ai-mathematical-olympiad-progress-prize-2. Kaggle.

Kuofeng Gao, Huanqia Cai, Qingyao Shuai, Dihong Gong, and Zhifeng Li. 2024. Embedding self-correction as an inherent ability in large language models for enhanced mathematical reasoning. *arXiv preprint arXiv:2410.10735*.

Mauricio J Giuliodori, Heidi L Lujan, and Stephen E DiCarlo. 2006. Peer instruction enhanced student performance on qualitative problem-solving questions. *Advances in physiology education*, 30(4):168–173.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.

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

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*.

Ke Ji, Jiahao Xu, Tian Liang, Qiuzhi Liu, Zhiwei He, Xingyu Chen, Xiaoyuan Liu, Zhijie Wang, Junying Chen, Benyou Wang, et al. 2025. The first few tokens are all you need: An efficient and effective unsupervised prefix fine-tuning method for reasoning models. *arXiv preprint arXiv:2503.02875*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Chengpeng Li, Mingfeng Xue, Zhenru Zhang, Jiaxi Yang, Beichen Zhang, Xiang Wang, Bowen Yu, Binyuan Hui, Junyang Lin, and Dayiheng Liu. 2025. Start: Self-taught reasoner with tools. *arXiv preprint arXiv:2503.04625*.

Dawei Li, Zhen Tan, Peijia Qian, Yifan Li, Kumar Satvik Chaudhary, Lijie Hu, and Jiayi Shen. 2024. Smoa: Improving multi-agent large language models with sparse mixture-of-agents. *arXiv preprint arXiv:2411.03284*.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 17889–17904.

Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. 2025. Zebralogic: On the scaling limits of llms for logical reasoning. *arXiv preprint arXiv:2502.01100*.

MAA. 2024. American Invitational Mathematics Examination - AIME 2024. https://maa.org/math-competitions/american-invitational-mathematics-examination-aime.

MAA. 2025. American Invitational Mathematics Examination - AIME 2025. https://maa.org/math-competitions/american-invitational-mathematics-examination-aime.

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

Sara Vera Marjanović, Arkil Patel, Vaibhav Adlakha, Milad Aghajohari, Parishad BehnamGhader, Mehar Bhatia, Aditi Khandelwal, Austin Kraft, Benno Krojer, Xing Han Lù, et al. 2025. Deepseek-r1 thoughtology: Let's think about llm reasoning. *arXiv preprint arXiv:2504.07128*.

OpenAI. 2024a. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/. Accessed: 2025-05-03.

OpenAI. 2024b. Learning to reason with llms. Accessed: 2025-05-03.

OpenAI. 2024c. Openai o1 system card. Accessed: 2025-05-03.

OpenAI. 2025a. gpt-oss model card (gpt-oss-120b & gpt-oss-20b). Accessed: 2025-11-01.

OpenAI. 2025b. Openai o3 mini. Accessed: 2025-05-03.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Julia J Snyder, B Elijah Carter, and Jason R Wiles. 2015. Implementation of the peer-led team-learning instructional model as a stopgap measure improves student achievement for students opting out of laboratory. *CBE—Life Sciences Education*, 14(1):ar2.

Qiushi Sun, Zhangyue Yin, Xiang Li, Zhiyong Wu, Xipeng Qiu, and Lingpeng Kong. 2023. Corex: Pushing the boundaries of complex reasoning through multi-model collaboration. *arXiv preprint arXiv:2310.00280*.

Yiliu Sun, Zicheng Zhao, Sheng Wan, and Chen Gong. 2025. Cortexdebate: Debating sparsely and equally for multi-agent debate. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9503–9523.

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. 2024. Mixture-of-agents enhances large language model capabilities. *arXiv preprint arXiv:2406.04692*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. *Preprint*, arXiv:2309.07597.

Zhenran Xu, Senbao Shi, Baotian Hu, Jindi Yu, Dongfang Li, Min Zhang, and Yuxiang Wu. 2023. Towards reasoning in large language models via multi-agent peer review collaboration. *arXiv preprint arXiv:2311.08152*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025a. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*.

Zhiyuan Zeng, Qinyuan Cheng, Zhangyue Yin, Yunhua Zhou, and Xipeng Qiu. 2025b. Revisiting the test-time scaling of o1-like models: Do they truly possess test-time scaling capabilities? *arXiv preprint arXiv:2502.12215*.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Zhihan Guo, Yufei Wang, Irwin King, Xue Liu, and Chen Ma. 2025. What, how, where, and how well? a survey on test-time scaling in large language models. *arXiv preprint arXiv:2503.24235*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Summarization Prompt Structure

Summary Trigger: (randomly select one)

- Alright, let's take a step back and summarize what we've figured out so far.
- Wait, let me quickly recap what I've concluded so far.
- Alright, let me shortly review the conclusions I've drawn so I can move forward more efficiently.
- Hmm, a quick summary of what I've figured out might help streamline the next part of my reasoning.
- Hold on, I should summarize the key points briefly to ensure I'm on the right track.
- Okay, before continuing, let me put together a brief summary of the insights I've gathered so far.
- Okay, time to consolidate everything I've found into a concise summary.

Summary Template: (randomly select one)

- In short, my current conclusions are that ...
- To summarize, based on my previous reasoning, I have currently found that ...
- In conclusion, the current key takeaways and results are ...
- In short, I've currently concluded that ...
- To summarize, my recent findings are ...
- In conclusion, the current insights and results I've gathered are ...

Figure 12: The structure of the summarization prompt used during LeaP.

A Prompts

Prompt for Different Tasks

GPQA:
Please show your choice in the answer field with only the choice letter, e.g.,"ANSWER": "C".
Math Tasks:
Please reason step by step, and put your final answer within \boxed.

Figure 13: Prompts for different tasks.

Prompt for Mixture-of-Agents

Problem: {problem}
You have been provided with a set of responses from various open-source models to the latest user query. Your task is to synthesize these responses into a single, high-quality response. It is crucial to critically evaluate the information provided in these responses, recognizing that some of it may be biased or incorrect. Your response should not simply replicate the given answers but should offer a refined, accurate, and comprehensive reply to the instruction. Ensure your response is well-structured, coherent, and adheres to the highest standards of accuracy and reliability.
Responses from models:

Figure 14: Prompts for mixture of agents.

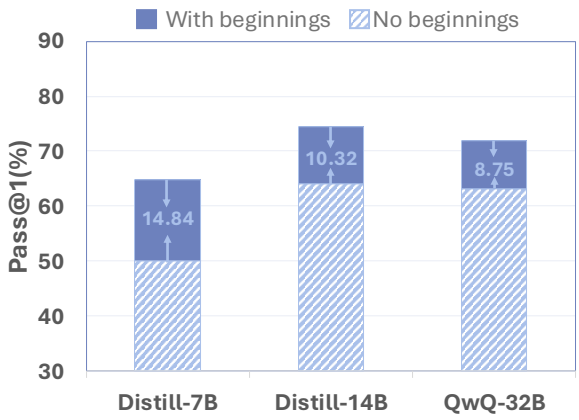
B Prefix Dominance Trap for Starting with Good Beginnings

While the "Prefix Dominance Trap" highlights how poor beginnings can severely constrain reasoning, a natural follow-up question is whether good beginnings are sufficient to ensure correct final answers. To this end, we conduct a follow-up experiment similar to those in Section 3.1 under the *good beginnings* setting, where the model start reasoning with a beginning of 15% average response length from correct reasoning paths.

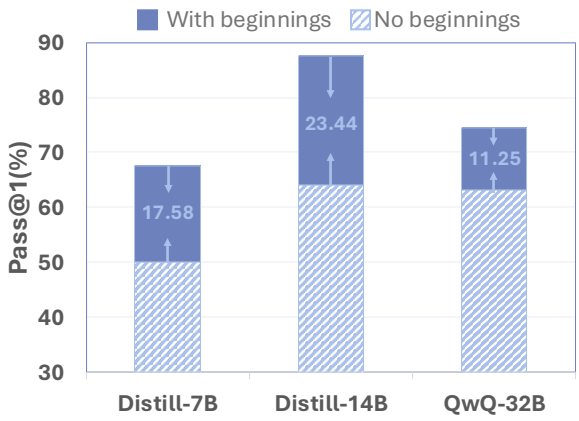
As shown in Figure 15a, models initialized with good beginnings indeed perform better than random or bad initializations. This suggests that early correct cues provide useful guidance. However, these beginnings do not fully eliminate reasoning errors—many responses still deviate in later steps and arrive at incorrect conclusions. This observation reinforces that good beginnings can reduce—but not entirely prevent—reasoning failures.

We then examine how LeaP perform in the same setting. As illustrated in Figure 15b, our method consistently surpasses independent reasoning, even when starting from already high-quality beginnings. For instance, DeepSeek-Distill-Qwen-14B improves from 74.38 to 87.50 in Pass@1. This

significant gain indicates that LeaP not only repairs faulty reasoning from poor prefixes but also mitigates subtle errors that emerge even when the reasoning starts correctly.



(a) Independent Reasoning



(b) LeaP

Figure 15: The results of starting with a good beginning of 15% average response length.

C Cons@32 Results

To provide a comprehensive evaluation, we also report the Cons@N results in Table 4. We combine the AIME 2024 (MAA, 2024), AIME 2025 (MAA, 2025), and AIMO 2025 (Frieder et al., 2024) benchmarks into a single category, denoted as *Math Comb.*, due to the small size of the individual math benchmarks. In general, the results under the selected two routing strategies consistently surpass the two baselines by a large margin. Specifically, on GPQA (Rein et al., 2024), the Top-4 *Hybrid* setting achieves a 10.10 improvement over the Independent Reasoning baseline. On the average of the three math benchmarks, the Top-2 *Hybrid* setting outperforms the baseline by 4.29 on QwQ-32B (Team, 2025).

D Case Study

D.1 Case Study on Comparing LeaP and Baseline with Bad Beginning

Case Study on Baseline with Bad Beginning

Problem:
Let A, B, C, and D be point on the hyperbola $\frac{x^2}{20} - \frac{y^2}{24} = 1$ such that $ABCD$ is a rhombus whose diagonals intersect at the origin. Find the greatest real number that is less than BD^2 for all such rhombi.

Model Response:
Okay, so I have this problem where there are four points A, B, C, D on the hyperbola given by the equation $\frac{x^2}{20} - \frac{y^2}{24} = 1$. ... So, we need to find the supremum (the least upper bound) of $BD^2 = 4(r^2 + s^2)$, such that the points satisfy the hyperbola equation and the condition $pr + qs = 0$. So, let me formalize this. Given: 1. $\frac{p^2}{20} - \frac{q^2}{24} = 1$. 2. $\frac{r^2}{20} - \frac{s^2}{24} = 1$. 3. $pr + qs = 0$. We need to find the supremum of $4(r^2 + s^2)$. **(Bad Beginning)**
...
Therefore, f(t) decreases from +infinity to 80 as t increases from 5/6 to infinity. Therefore, the infimum of BD^2 is 80, but it's never actually reached; it's a limit as t approaches infinity. Therefore, BD^2 can be made arbitrarily close to 80, but never less than 80. Therefore, the greatest real number less than BD^2 for all such rhombi is 80. Hence, the answer is $\boxed{80}$. **Final Answer** $\boxed{80}$

Figure 17: Case Study on baseline with bad beginning.

D.2 Case Study on Failure Examples

We provide some case studies to better illustrate our approach. In Figure 18, we present a failure case when applying LeaP to DeepSeek-R1-Distill-Qwen-7B. In this example, the model fails to clearly summarize its own reasoning, which results in the loss of partial reasoning information during summarization. This issue primarily relates to model size. Specifically, when the model is relatively small, incomplete summarization tends to occur more frequently.

In the second case, we demonstrate a situation where the model refuses to engage in communication. As shown in Figure 19, the model assumes it is alone and therefore ignores the perspectives of other peers, continuing its own reasoning trajectory. This case is observed in QwQ-32B but not in the distilled model from R1. We hypothesize that this difference may be related to the training

Table 4: The Cons@N results of Math benchmarks and GPQA Diamond.

Benchmarks	Models	Independent Reasoning	Dispersed		Hybrid	
			Top-2	Top-4	Top-2	Top-4
Math Comb. Cons@32	R1-7B	60.00	62.86	61.43	61.43	62.86
	R1-14B	68.57	68.57	70.00	70.00	71.43
	QwQ-32B	78.57	80.00	80.00	82.86	81.43
GPQA Diamond	R1-7B	50.00	59.60	58.59	56.57	60.10
	R1-14B	55.05	59.60	59.60	60.61	61.11
	QwQ-32B	60.61	67.68	63.64	63.13	68.18

method: QwQ-32B is trained via reinforcement learning (RL), which may lead the model to exhibit higher confidence.

D.3 Case Study on Success Examples

In the third case, we present a scenario where the model ultimately reaches the correct answer through interaction with its peers. As shown in Figure 20, the model initially fails to fully summarize its own reasoning process, leading to the loss of partial reasoning information during summarization. However, during the subsequent interaction with peers, the model receives a clearer and more explicit reasoning path provided by them. By verifying the summaries generated by other peers, the model gradually identifies the correct solution to the problem.

In the fourth case, we present a scenario where the model initially generates an incorrect answer during the reasoning process. Meanwhile, other peers also produce flawed summaries, though some provide correct ones. Through interaction and mutual verification, the model eventually identifies the correct reasoning path. As shown in Figure 21, the model first summarizes its own reasoning and arrives at the incorrect answer of 389. Peer 1 and Peer 2 provide incorrect summaries, while only Peer 3 offers a reasonable one. By verifying the summaries from all peers, the model determines that the summaries from Peer 1 and Peer 2 are inconsistent. In contrast, the summary from Peer 3 is considered reliable. Based on this verification process, the model gradually uncovers the correct solution to the problem.

E Training Details of LeaP-T

We utilize 8 NVIDIA A100 GPUs to train and evaluate our models. We use LLama-Factory (Zheng et al., 2024) to train our models on AIME 1984 to 2023. [1] The training hyperparameters are as follows:

F Comparison with MoA and Self-Correct Method

We also test MoA (Wang et al., 2024) as a baseline under the setting of 4 Layers and 3 Agents, with the prompts available in Appendix A. As shown in Table 6, MoA costs approximately 80K tokens per problem, but the results show that it does not transfer well to LRMs. The reason is that LRMs cannot follow the user's instructions effectively. We find that even when the correct answer is achieved in the intermediate layers, the model still reaches incorrect conclusions in the final layer.

G Evaluation Results of LeaP in Single Communication on Four Reasoning Benchmarks

We evaluate a simplified LeaP variant (LeaP-S) that retains only the first LeaP block at T tokens and continues generation along a single path. This mirrors how humans often work: gathering ideas early, then independently verifying them. As shown in Table 8, this single communication variant outperforms DeepSeek-R1-Distill-Qwen-14B. For example, on AIME 2024, it improves Pass@1 from 64.47 to 68.13; on GPQA Diamond, from 53.47 to 57.42. Even with just one block, it achieves a higher average Pass@1 (55.95 vs. 53.86), showing that early peer exposure effectively guides reasoning. Gains are most notable in math and multi-hop tasks, where early external signals reduce error accumulation. This variant also balances performance and efficiency, making it suitable for single-path scenarios. We further explore different hyperparameters on AIME 2024 (Table 9). Increasing

[1] LLama-Factory is licensed under the Apache-2.0 license. All data from AIME is licensed under the MIT license.

Table 5: Hyperparameters for LeaP-T training.

Hyperparameter	1.5B Model	7B Model	14B Model
Batch size	16	16	16
Learning rate	1×10^{-5}	1×10^{-5}	1×10^{-5}
Learning rate scheduler	Cosine decay	Cosine decay	Cosine decay
Warmup ratio	0.05	0.05	0.05
Optimizer	AdamW	AdamW	AdamW
Weight decay	1×10^{-4}	1×10^{-4}	1×10^{-4}
Max sequence length	16K	16K	16K
Training epochs	8.0	8.0	5.0
Precision	bfloat16	bfloat16	bfloat16

Table 6: Performance vs. compute cost comparison under 7B scale.

Method	Score (%)	Compute (Avg. Tokens / Q)
R1-7B	66.67	∼74,000
MoA + R1-7B	53.33	∼77,975
LeaP + R1-7B	71.67	∼69,860
LeaP-T 7B	73.33	∼67,998

the interval T from 2K to 4K improves accuracy (e.g., 67.92 to 69.48 at $k = 8$), and using more peers (e.g., $k = 8$ vs. $k = 2$ at $T = 4$K) also yields better results.

Table 8: The Pass@1 results of independent reasoning and LeaP-S on AIME 2024, AIME 2025, AIMO 2025 and GPQA Diamond for 14B model.

Benchmarks	R1-14B	$T = 2$**K**	$T = 4$**K**
AIME 2024	64.47	66.25	**68.13**
AIMO 2025	46.87	49.38	**50.94**
AIME 2025	**48.64**	45.83	47.29
GPQA Diamond	53.47	55.11	**57.42**
Avg.	53.86	54.14	**55.95**

H Other In-depth Analysis

H.1 On Computational Overhead

We next analyze the computational overhead of LeaP. The additional cost comes from processing peer summaries at communication turns. Importantly, while these summaries add tokens to the context, they are processed in the parallelized prefill stage, whereas the dominant latency factor is the serial autoregressive generation.

H.1.1 FLOPs

To analyze the FLOPs overhead, we first define the key architectural parameters and derive the FLOPs for a standard forward pass.

Symbol	Definition
d_{model}	Model hidden dimension (hidden_size)
d_{ff}	Feed-forward intermediate dimension
n_{layers}	Total number of layers
n_{heads}	Number of attention heads for Query
n_{kv_heads}	Number of attention heads for Key/Value (GQA)
d_{head}	Dimension of each attention head
v_{size}	Vocabulary size
N	Input sequence length

Table 10: Notation for FLOPs analysis.

The FLOPs for processing a sequence of length N can be broken down as follows.

To simplify notation, let $d_{\text{model}} = d_{\text{model}}$, $d_{\text{ff}} = d_{\text{ff}}$, $n_{\text{kv_heads}} = n_{\text{kv_heads}}$, and $d_{\text{head}} = d_{\text{head}}$.

Per-layer FFN FLOPs.

$$\text{FLOPs}_{\text{FFN}} = 4Nd_{\text{model}}d_{\text{ff}} \tag{2}$$

Per-layer Self-Attention FLOPs.

$$\begin{aligned}\text{FLOPs}_{\text{Attn}} &= 2Nd_{\text{model}}^2 + 4Nd_{\text{model}}n_{\text{kv_heads}}d_{\text{head}} \\ &\quad + 2N^2d_{\text{model}} \\ &\quad + 2N^2d_{\text{model}} + 2Nd_{\text{model}}^2 \\ &= 4N^2d_{\text{model}} + 4Nd_{\text{model}}^2 \\ &\quad + 4Nd_{\text{model}}n_{\text{kv_heads}}d_{\text{head}}\end{aligned} \tag{3}$$

Total FLOPs. Let $D = d_{\text{ff}} + d_{\text{model}} + n_{\text{kv_heads}}d_{\text{head}}$. Then

$$\begin{aligned}\text{FLOPs}(N) &= n_{\text{layers}}(\text{FLOPs}_{\text{FFN}} + \text{FLOPs}_{\text{Attn}}) \\ &\quad + 2Nd_{\text{model}}v_{\text{size}} \\ &= n_{\text{layers}}(4N^2d_{\text{model}} \\ &\quad + 4Nd_{\text{model}}D) + 2Nd_{\text{model}}v_{\text{size}}.\end{aligned} \tag{4}$$

Table 7: Performance comparison between LeaP and CoSC.

Benchmark	Method	Pass@1	Pass@2	Pass@4	Pass@6	Pass@8
AIME 2024	CoSC	0.5312	0.6533	0.7533	0.7667	0.7933
	LeaP	0.6052	0.7133	0.7433	0.7767	0.8000
AIME 2025	CoSC	0.3833	0.4667	0.5400	0.5667	0.5933
	LeaP	0.3844	0.4933	0.5667	0.5733	0.6033
AIMO	CoSC	0.3969	0.5000	0.5400	0.5667	0.5933
	LeaP	0.4500	0.5000	0.5800	0.5600	0.6000

Table 9: The Pass@1 results for varying k values and position of LeaP block on AIME 2024 with the 14B model in LeaP-S.

	$k=2$	$k=4$	$k=6$	$k=8$	$k=16$ (Top-2)	$k=16$ (Top-4)
$T=2$k	65.21	66.25	65.73	67.92	66.46	66.86
$T=4$k	65.73	68.13	68.85	69.48	67.92	69.27

For simplicity we approximate the dominant terms as:

$$\text{FLOPs}(N) \approx 4\, n_{\text{layers}} N d_{\text{model}} (N + d_{\text{ff}} + d_{\text{model}}). \tag{5}$$

The extra FLOPs from LeaP occur during communication turns. At each turn i (at context length iT), the model processes new summary tokens. The cost of adding $L_{\text{new}} = (k+1)L_{\text{sum}}$ tokens to a context of length M is:

$$\Delta\text{FLOPs}(M, L_{\text{new}}) = \text{FLOPs}(M + L_{\text{new}}) - \text{FLOPs}(M). \tag{6}$$

Example: For R1-7B with $T = 4096$, $k = 4$, $L_{\text{sum}} = 143$ ($L_{\text{new}} \approx 715$), and $n_{\text{turns}} = 1.41$:

$$\begin{aligned}\text{FLOPs}_{\text{Overhead}} \approx\ & \big[\text{FLOPs}(4096 + 715) \\ & - \text{FLOPs}(4096)\big] \\ & + 0.41\big[\text{FLOPs}(8192 + 1430) \\ & - \text{FLOPs}(8192 + 715)\big].\end{aligned} \tag{7}$$

The baseline generates on average 9,045 tokens:

$$\text{FLOPs}_{\text{Baseline}} \approx \text{FLOPs}(9045). \tag{8}$$

Therefore, the relative overhead is:

$$\text{Overhead (\%)} = \frac{\text{FLOPs}_{\text{Overhead}}}{\text{FLOPs}_{\text{Baseline}}} \times 100\% \approx 12.08\%. \tag{9}$$

This confirms that while LeaP introduces a measurable FLOPs cost at each communication step, this overhead is moderate and is offset by the shorter reasoning paths it enables, thereby reducing the total number of decode steps.

H.1.2 Total Tokens

We analyze LeaP's inference efficiency based on the total number of tokens and the frequency of self-correction, often termed "aha" moments. As illustrated in Figure 22, LeaP does not generate significantly more tokens than the baseline across the QwQ-32B, DeepSeek-R1-Distill-Qwen-7B, and DeepSeek-R1-Distill-Qwen-14B models. In fact, for certain benchmarks like GPQA Diamond, its average response length is sometimes shorter.

We further analyze "aha" moments using keywords from previous studies (Zeng et al., 2025a; Li et al., 2025; Chen et al., 2025). The results consistently show that LeaP exhibits fewer "aha" moments than the baseline. For example, with QwQ-32B, LeaP has 16.4% fewer such instances across three math benchmarks. Additionally, we observe that the number of "aha" moments for Top-4 settings is consistently lower than for Top-2 settings. These findings indicate that receiving peer opinions and results reduces the model's need for reflection and self-correction, leading to a more streamlined reasoning process.

H.2 On Generalization to Non-math/STEM Tasks

To test LeaP's generalization beyond mathematical and scientific reasoning, we evaluated it on ZebraLogic (Lin et al., 2025), a benchmark of over 3,000 natural-language logical puzzles. Following the evaluation protocol in Section 4.1, we set LeaP with $T = 4096$, a Top-4 Dispersed Router, and report Pass@1 and Cons@8 to capture single-path quality and ensemble consensus.

Table 11: LeaP performance on ZebraLogic.

Model	Pass@1	Cons@8
R1-7B	46.88	53.11
LeaP + R1-7B	**53.08**	**76.43**
R1-1.5B	30.20	42.07
LeaP + R1-1.5B	**37.45**	**62.04**

These results in table 11 show that LeaP substantially improves both single-path accuracy (Pass@1) and ensemble consensus (Cons@8) on non-math logical reasoning tasks. This provides strong empirical evidence that the benefit of structured peer insights is not limited to numerical or STEM reasoning.

H.3 On Homogenization

We acknowledge that homogenization is a factor in designing the LeaP framework, our analysis in Section 6.1.3 and Section 6.1.4 shows that earlier communication between peers leads to more similar results. However, as the main experiments indicate, the overall benefit of communication, especially when it doesn't occur too early, significantly outweighs the risk of this convergence.

An insightful hypothesis is that the diversity in our summary prompting format (as shown in Figure 12) helps mitigate this homogenization. To test this, we conducted an ablation comparing our default setting (random prompts from Figure 12) with a fixed trigger and template. The fixed prompt was:

> **Summary Trigger:** *Alright, let's take a step back and summarize what we've figured out so far.*
> **Summary Template:** *In short, my current conclusions are that*

The results in Table 12 show only minor performance differences between using random and fixed prompts. This suggests that while prompt diversity may contribute slightly, it is **not the decisive factor** in preventing homogenization or in LeaP's overall success. This finding reinforces our main conclusion: **the core benefit comes from the peer-learning mechanism itself**, and this benefit significantly outweighs the risk of convergence, especially when communication is timed appropriately.

H.4 On Design of Routers

H.4.1 Necessity of the Router (Router vs. Broadcast)

To validate the necessity of a selective routing mechanism, we conducted an ablation study comparing our proposed router against a "broadcast" baseline. In the broadcast setting, every reasoning path receives summaries from all other paths, which is equivalent to setting the number of peers k to the total number of paths ($k = 32$ in our experiments). This setup effectively disables selective communication. All other experimental parameters were held constant to ensure a fair comparison.

The results, presented in Table 13, show that our selective routing approach (Top-4) consistently outperforms the full broadcast baseline across all benchmarks. This finding suggests that merely increasing the volume of information is suboptimal and can be detrimental to performance.

This conclusion is further corroborated by our analysis of communication traffic in Section 6.1.2 (Figure 7), where we demonstrate that performance does not increase monotonically with the number of peer summaries (k). Instead, a clear trade-off emerges:

- **Too few summaries** ($k < 4$) provide insufficient information for robust self-correction and refinement.
- **Too many summaries** ($k > 4$) can introduce distracting or contradictory information, overwhelming the model's reasoning capabilities and leading to cognitive overload.

Our experiments identify $k = 4$ as the optimal configuration, striking a balance between providing rich, diverse insights and avoiding information overload.

H.4.2 Semantic-Based Router

To explore more advanced routing strategies, we conducted an experiment using semantic similarity as the basis for routing, rather than the heuristic-based Levenshtein distance. For this experiment, we employed the `bge-large-en-v1.5` (Xiao et al., 2024) model to compute embeddings for each summary. We then applied the same Dispersed Routing strategy, selecting the Top-4 most dissimilar summaries based on cosine similarity. All other settings were consistent with those described in Section 3.2.

As shown in Table 14, routing based on semantic dissimilarity yields a consistent, albeit modest, per-

Table 12: Ablation study on the impact of prompt diversity on homogenization. Performance differences are minimal, suggesting prompt diversity is not the primary factor.

Pass@1 (R1-7B)	Random Prompt	Fixed Prompt
AIME 2024	60.52	60.21
AIME 2025	38.44	40.48
AIMO 2025	45.00	40.31

Table 13: Comparison of Pass@1 performance between our selective router ($k = 4$) and a broadcast baseline ($k = 32$). Selective routing consistently yields better results.

Benchmark	Pass@1 (w/ Router, $k = 4$)	Pass@1 (Broadcast, $k = 32$)
AIME 2024	60.52	59.03
AIME 2025	38.44	37.71
AIMO 2025	45.00	41.88

formance improvement. This result validates the intuition that a deeper, meaning-based understanding of peer summaries can lead to more effective information exchange. However, this approach introduces the computational overhead of encoding the summaries. We believe this points toward a promising future direction where peer-learning paradigms are integrated directly into the model architecture. Such an integration could enable routing based on internal feature similarity, which might prove to be both more effective and more computationally efficient than methods relying on explicit natural language summaries.

H.5 Human Evaluation on The 11th Problem of AIME 2024

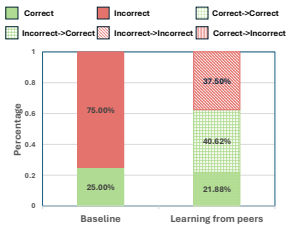


Figure 23: Human evaluation on the 11th problem of AIME 2024.

While benchmarks confirm LeaP's improved reasoning, they may not reveal underlying behavioral changes. For deeper insight, we performed a human evaluation comparing QwQ-32B's baseline and LeaP outputs on the 11th AIME 2024 problem. Figure 23 shows that on this problem, LeaP significantly increased correct responses from 25.00% (8/32) to 62.50% (20/32). We categorized response transitions to understand this improvement: *Correct* $\rightarrow$ *Correct*, *Incorrect* $\rightarrow$ *Correct*, *Incorrect* $\rightarrow$ *Incorrect*, and *Correct* $\rightarrow$ *Incorrect*. Notably, 40.62% of responses were *Incorrect* $\rightarrow$ *Correct*, showing LeaP repairs many flawed paths with peer communication. Importantly, no responses transitioned from *Correct* $\rightarrow$ *Incorrect*, suggesting communication rarely disrupts correct reasoning but primarily fixes errors. Case studies are in Appendix D.3.

H.6 Other Sensitivity Analysis

H.6.1 On Temperature

We conduct an in-depth analysis of the temperature parameter τ by varying it from 0.1 to 1.0, while keeping all other settings fixed. We use Top-4 Dispersed routing and set the communication interval to $T = 4$k tokens. As shown in Figure 24, when the temperature is low ($\tau \leq 0.3$), the model achieves an average accuracy of around 71, similar to the *Clustered* router baseline. This is expected, as a low temperature reduces output diversity, thereby limiting the benefits of peer communication. On the other hand, when the temperature is too high (e.g., $\tau \geq 0.9$), performance drops noticeably. This is likely due to excessive randomness, which may lead the model away from coherent reasoning or instruction following. Overall, a moderate temperature appears to strike a good balance between diversity and stability.

Table 14: Performance comparison between the heuristic-based (Levenshtein) router and a semantic-based (embedding similarity) router. The semantic approach shows modest but consistent improvements.

Benchmark	Heuristic-based	Semantic-based
AIME 2024	60.52	61.56
AIME 2025	38.44	39.48
AIMO 2025	45.00	45.21

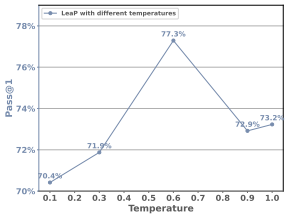
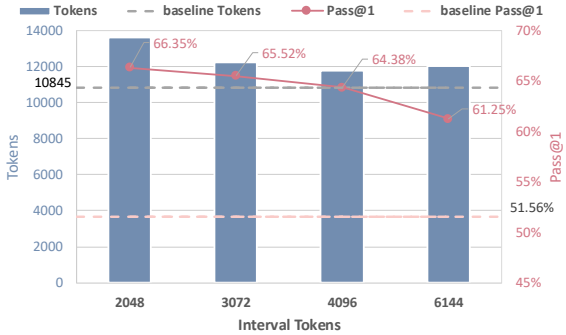
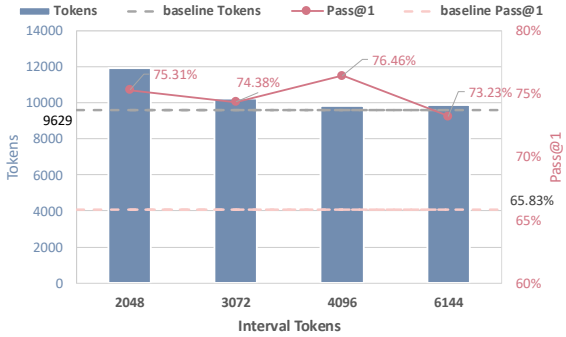


Figure 24: Pass@1 results of various temperatures on AIME 2024.



(a) 7B



(b) 14B

Figure 25: Pass@1 and total tokens on AIME 2024 for LeaP-T-7B and LeaP-T-14B models, evaluated across interval tokens of LeaP from 2048 to 6144.

H.6.2 On Communication Granularity T for LeaP-T

To further study the effect of communication granularity, we analyze LeaP-T under different token intervals T. As shown in Figure 25a, for the 7B version, the Pass@1 score on AIME 2024 decreases from 66.35 to 61.25 as T increases, while the number of generated tokens drops accordingly. This trend is consistent with the analysis in Section 6.1.1: more frequent communication (i.e., smaller T) improves performance slightly, but increases token consumption due to more frequent summarization and message exchange. We observe a similar trend for the 14B version in Figure 25b.

H.6.3 On Communication Position for LeaP-T

Furthermore, Figure 26 analyzes when communication is most effective. The results indicate that performance peaks when the LeaP block is inserted at 4K tokens. Specifically, Pass@1 increases from 68.85 at 2K to 71.77, then declines to 69.69 at later positions. Although a slight recovery to 71.25 is observed, the overall pattern suggests that earlier communication tends to yield better results. These results are consistent with our analysis in Section 6.1.4, and highlight the importance of timely information exchange in improving performance.

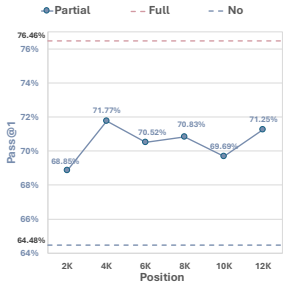


Figure 26: Performance of partial LeaP on the 14B model with LeaP-T, evaluated across the positions of LeaP block ranging from 2K to 12K tokens

H.7 Robustness at Various Difficulty Levels for LeaP-T

We also report the performance of LeaP-T across the five difficulty levels of AIME 2024. These difficulty levels follow the same categorization introduced in Section 6.2.2, based on model (without peer reasoning) accuracy. To exclude the impact of distilling with a 32B model, we conduct distillation using SFT without LeaP, which serves as our baseline. This setup allows for a direct comparison with our LeaP-T approach. Across all five difficulty levels, LeaP-T generally outperforms the baseline. For instance, as shown in Figure 29, LeaP-T achieves higher accuracy while consuming fewer tokens. These findings are consistent with the analysis presented in Section 6.2.2. This is because LeaP facilitates earlier consensus during the reasoning process, thereby reducing unnecessary computational overhead caused by overthinking. Notably, the improvement of LeaP-T over the baseline suggests that its performance gain is not solely attributed to knowledge transfer through distillation. Instead, it underscores the effectiveness of a training paradigm centered on *learning from peers*.

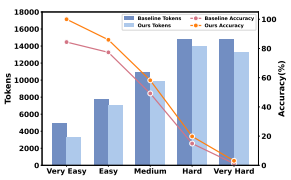


Figure 27: The Pass@1 and average token distribution across different difficulty levels, from Very Easy to Very Hard, for the LeaP-T-1.5B.

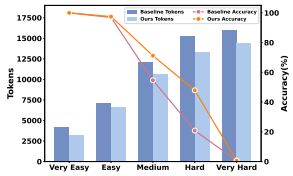


Figure 28: The Pass@1 and average token distribution across different difficulty levels, from Very Easy to Very Hard, for the LeaP-T-7B.

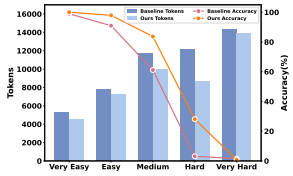


Figure 29: The Pass@1 and average token distribution across different difficulty levels, from Very Easy to Very Hard, for the LeaP-T-14B.

H.8 Behavior Difference of LeaP using RL-*ed* and Non-RL models

We observe that reinforcement learning (RL) models, such as QwQ, display higher confidence in their reasoning. This is particularly evident in the increasing proportion of *Unaffected* cases at later stages of reasoning. The reinforcement learning process tends to encourage more consistent behavior, with certain tokens' sampling probabilities increasing, guiding the model back to familiar reasoning paths. This aligns with findings from previous studies (Ji et al., 2025; Yu et al., 2025), which show that RL-trained models tend to prefer their learned reasoning strategies, exhibiting greater self-assurance in their conclusions. This confidence may explain the model's resistance to altering its reasoning path when provided with peer insights, especially in the later stages of reasoning.

Case Study on LeaP with Bad Beginning

Problem:
Let A, B, C, and D be point on the hyperbola $\frac{x^2}{20} - \frac{y^2}{24} = 1$ such that $ABCD$ is a rhombus whose diagonals intersect at the origin. Find the greatest real number that is less than BD^2 for all such rhombi.

Model Response: Okay, so I have this problem where there are four points A, B, C, D on the hyperbola given by the equation $\frac{x^2}{20} - \frac{y^2}{24} = 1$. ... So, we need to find the supremum (the least upper bound) of $BD^2 = 4(r^2 + s^2)$, such that the points satisfy the hyperbola equation and the condition $pr + qs = 0$. So, let me formalize this. Given: 1. $\frac{p^2}{20} - \frac{q^2}{24} = 1$. 2. $\frac{r^2}{20} - \frac{s^2}{24} = 1$. 3. $pr + qs = 0$. We need to find the supremum of $4(r^2 + s^2)$. **(Bad Beginning)**
...
<summarize> In conclusion, the current key takeaways and results are: ... Using hyperbola parametrization and vector properties, I derived an expression for BD² in terms of a parameter b, which is tanh u. The expression for BD² is $\frac{80+96b^2}{1-b^2}$ with the constraint $|b| \in (\frac{5}{6}, 1)$. By analyzing the derivative of this function, I found that it's increasing in the interval, implying the minimum BD² occurs as b approaches $\frac{5}{6}$. Therefore, the minimal BD² is achieved in the limit as b approaches $\frac{5}{6}$. </summarize> **(Incorrect Summary)**
<peer_summaries>
Peer 1: "To summarize, based on my previous reasoning, I have currently found that BD^2 can be expressed as $80 + (22v)/3$, where $v = s^2$. The variable v is constrained by the equation $u = [600(v - 24)]/(11v - 600)$, where $u = q^2$. The variable v must be either less than 24 or greater than approximately 54.545...
Peer 2: "In short, I've currently concluded that the expression for BD^2 in terms of $u = p^2$ is $480 \times \frac{11u-120}{11u-720}$. I also noticed that as u approaches $\frac{720}{11}$ from above, BD^2 tends to infinity, which seems contradictory because points on a hyperbola shouldn't allow for infinitely long diagonals...
Peer 3: "To summarize, my recent findings are: I attempted to express p and s in terms of q and r using the condition $pr + qs = 0$, leading to $p = kq$ and $s = -kr$...
Peer 4: "In conclusion, the current insights and results I've gathered are: Expressed BD² in terms of t (where t = q²), transformed the equation into a function f(t) = 80 + [4400(t + 24)] / (11t - 600), and simplified it to f(t) = 480 + (345600/11)/(t - 600/11). The domain of t is t > 600/11. So, as t approaches infinity, f(t) approaches 480, and as t approaches 600/11 from the right, f(t) approaches infinity. Therefore, BD² can be made arbitrarily large, but the problem asks for the greatest real number less than BD² for all such rhombi. So, that number is 480." **(Correct Summary from Peers)** </peer_summaries>
...
From the comments, it appears that some peers have approached the problem differently and arrived at different expressions for BD².Notably, Peer 4 suggests that BD² approaches 480 as t approaches infinity, and that BD² can be made arbitrarily large, but the greatest number less than BD² for all rhombi is 480. **(Self-Verification)** Wait, that seems conflicting with my earlier conclusion where BD² approaches infinity as b approaches 1. So, perhaps I made a mistake in my parametrization. Wait, let me check Peer 4's approach. Peer 4 says: Expressed BD² in terms of t (where t = q²)
...
Therefore, the infimum of BD² is 480, so the greatest real number less than BD² for all rhombi is 480. But in my own reasoning, I thought BD² can be made as large as desired, which would suggest that the infimum is 480, and the supremum is infinity. So, the greatest real number less than BD² for all rhombi is 480...
Therefore, the answer is 480. But wait, in my earlier parametrization, I had BD² tending to infinity as b approaches 1, but here, it's showing that BD² can be made as small as 480 and as large as infinity. Therefore, the greatest real number less than BD² for all rhombi is 480. So, the answer is 480. **Final Answer** $\boxed{480}$

Figure 16: Case Study on LeaP with bad beginning.

Case Study on Summarization Failure

Problem:
Every morning Aya goes for a 9-kilometer-long walk and stops at a coffee shop afterwards. When she walks at a constant speed of s kilometers per hour, the walk takes her 4 hours, including t minutes spent in the coffee shop. When she walks $s+2$ kilometers per hour, the walk takes her 2 hours and 24 minutes, including t minutes spent in the coffee shop. Suppose Aya walks at $s+\frac{1}{2}$ kilometers per hour. Find the number of minutes the walk takes her, including the t minutes spent in the coffee shop.

Model Response:
...
Wait, wait, from $x^{253}-34x+33=0$, $x=k^8$, so $x=1$ is a root, so we can factor it as $(x-1)(something)=0$. But since $x=1$ is a root, and we're interested in the other root, perhaps it's a very small x, but I'm not sure. Wait, perhaps I can write the equation as $x^{253}=34x-33$. Hold on, I should summarize the key points briefly to ensure I'm on the right track. <summarize> To summarize, my recent findings are: The problem involves two circles of radius 34 and 2024 arranged in a triangle ABC, with each circle tangent to two sides and the previous circle. The inradius R of ABC can be expressed as $R=34\cdot\frac{1-k^8}{1-k}$ for the 34-radius circles, and $R=\frac{1-k^{2024}}{1-k}$ for the 1-radius circles. Setting these equal gives the equation $34\cdot(1-k^8)=1-k^{2024}$, which simplifies to $k^{2024}-34k^8+33=0$. Letting $x=k^8$, this becomes $x^{253}-34x+33=0$, which factors as $(x-1)(x^{252}+x^{251}+\cdots+x+1)-34x+33=0$. Since $x=1$ is a root, but we need another real root where $x<1$. Approximating, $x\approx 0.970588$, leading to $k\approx 0.9963$. Then, $R\approx$ ... (**truncated at 256 tokens**) </summarize>
...

Figure 18: The case study about the reasoning model fail to summarize concisely.

Case Study on Reflection Failure

Problem:
Torus T is the surface produced by revolving a circle with radius 3 around an axis in the plane of the circle that is a distance 6 from the center of the circle (so like a donut). Let S be a sphere with a radius 11. When T rests on the inside of S, it is internally tangent to S along a circle with radius r_i, and when T rests on the outside of S, it is externally tangent to S along a circle with radius r_o. The difference r_i-r_o can be written as $\frac{m}{n}$, where m and n are relatively prime positive integers. Find $m+n$. [asy] unitsize(0.3 inch); draw(ellipse((0,0), 3, 1.75)); draw((-1.2,0.1)..(-0.8,-0.03)..(-0.4,-0.11)..(0,-0.15)..(0.4,-0.11)..(0.8,-0.03)..(1.2,0.1)); draw((-1,0.04)..(-0.5,0.12)..(0,0.16)..(0.5,0.12)..(1,0.04)); draw((0,2.4)–(0,-0.15)); draw((0,-0.15)–(0,-1.75), dashed); draw((0,-1.75)–(0,-2.25)); draw(ellipse((2,0), 1, 0.9)); draw((2.03,-0.02)–(2.9,-0.4)); [/asy]

Model Response:
...
<peer_summaries>
...
Peer 4: "In conclusion, the current insights and results I've gathered are: By setting up the equations for the sphere and the torus, and analyzing the gradients for tangency, I derived that at points of tangency, the z-component gives $k=2$ (assuming $z\neq 0$). This led to an equation relating the radial distance $\rho=sqrt(x^2+y^2)$ to the major radius $R=6$. Solving this gives a specific ρ value, which when substituted back into the equations, yields a valid solution for z. This allows computing the radii r_i and r_o for internal and external tangency, leading to the desired difference. However, I must ensure that the configuration (internal vs. external) corresponds correctly to the direction of the normals." </peer_summaries> Hmm, it seems that my peers have given me some comments, so let me check if anyone's conclusions are different from mine before I continue my own reasoning. But since I'm alone, I'll proceed. Let me try to proceed with the standard equation of the torus. Let me define the torus such that its $axis$ is the $x-axis$. The generating circle has a major radius $R=6$, so its center is at (6, 0, 0), and minor radius $r=3$. The equation of the torus is: $(sqrt(y^2+z^2))^2+(x-6)^2=3^2$? Wait, no, perhaps I need to think differently. Let me recall that the standard equation for a torus rotated around the $x-axis$ with major radius R and minor radius r is: $(sqrt((sqrt(y^2+z^2)-R)^2+x^2))$? Wait, perhaps I should use a different approach.
...

Figure 19: The case study that the QwQ-32B reject to understand the summaries from peers.

Case Study on LeaP Success

Problem:
"Each vertex of a regular octagon is independently colored either red or blue with equal probability. The probability that the octagon can then be rotated so that all of the blue vertices end up at positions where there were originally red vertices is $\frac{m}{n}$, where m and n are relatively prime positive integers. What is $m+n$?"

Model Response:
...
<peer_summaries>
...
<summarize> In conclusion, the current insights and results I've gathered are as follows: - **Good Subsets**: Subsets with 0 to 3 elements are all good, totaling $1 + 8 + 28 + 56 = 93$ subsets. Subsets of size 4 may include some bad subsets, while subsets of size 5–8 are mostly bad, though not all. - **Bad Subsets of Size 4**: Examples like 0,1,3,4 cannot be rotated to be disjoint, indicating that some size-4 subsets are bad. Calculating their exact count requires identifying patterns or symmetries that make these subsets resistant to rotation. - **Symmetry and Complementarity**: The problem's symmetry suggests that the number of bad subsets of size k equals those of size $8 - k$. However, since subsets of size 3 (complement of 5) are all good, subsets of size 5 must also be good, contradicting earlier assumptions.This implies a need to reevaluate. - **Reevaluation**: Perhaps subsets of size 5 are actually good, and the real bad subsets are only those of size 7, 8, and some specific size 4 subsets. This would mean the total bad subsets are 1 (... </summarize> (**truncated at 256 tokens**) <peer_summaries> Peer 1: "..."Peer 2:"..."Peer 3: "..."Peer 4:"To summarize, my recent findings are as follows: - **Subsets of Size 4**: - **Rotation by 1 or 7**: Only two valid subsets (checkerboard patterns 0,2,4,6 and 1,3,5,7). - **Rotation by 2 or 6**: Requires selecting opposite pairs in each of the two 4-cycles formed by the rotation. This yields $2 \times 2 = 4$ subsets. - **Rotation by 4**: Selecting one vertex from each of the four diametrical pairs, resulting in $2^4 = 16$ subsets. - **Overlap Considerations**: No overlaps between these categories, as the subsets from different rotations are structurally distinct. - **Total Valid Size 4 Subsets**: 2 +4 +16 =22. - **Total Favorable Subsets**: - Size 0:1 - Size 1:8 - Size 2:28 - Size 3:56 - Size 4:22 - **Total**:1+8+28+56+22 =115 - **Probability Calculation**: - Total subsets:..."</peer_summaries> Hmm, it seems that my peers have given me some comments, so let me check if anyone's conclusions are different from mine before I continue my own reasoning. Peer 1 is talking about independent sets on a cycle with distance constraints. Peer 2 suggests that size4 subsets can be counted by analyzing different rotation types, leading to 22 valid subsets. Peer 3 mentions inclusion-exclusion over rotation classes. Peer 4 calculates total favorable subsets as 115, leading to probability 115/256. Wait, Peer4's calculation: Size0:1 Size1:8 Size2:28 Size3:56 Size4:22 (only those that are good) Total: 1+8+28+56+22 =115 Thus probability 115/256, which simplifies to 115 and 256 are coprime? 256 is 2^8. 115 is 5×23, so yes. Thus m+n=115+256=371. But I need to verify if this is correct. According to Peer4's reasoning: For size4 subsets, the valid (good) subsets are 22. The bad subsets of size4 are 70-22=48. Therefore, total bad subsets are: Size5:56 Size6:28 Size7:8 Size8:1 Size4 bad:48 Total bad:56+28+8+1+48=141 Thus total good subsets:256-141=115 Therefore, probability 115/256 Thus the answer is 115+256=371. But I need to verify if Peer4's count of size4 subsets is correct. Peer4 says
...
Yes. Thus, these subsets are good. The checkerboard patterns are also good. Therefore, the count of22 valid subsets of size4 is correct. Thus, the total good subsets are115, leading to the probability of115/256. Thus, the final answer is115 +256= 371.(**Final Answer** $\boxed{371}$)

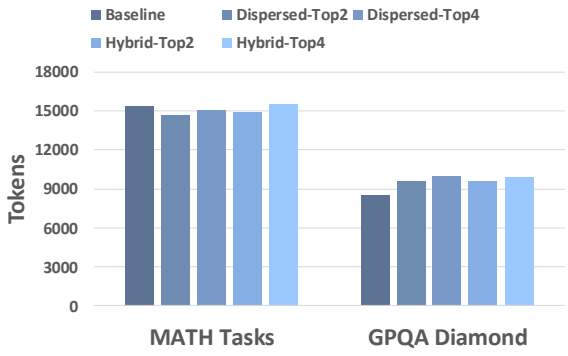
Figure 20: The case study that the model learn from peers successfully, which lead to a correct final answer.

Case Study on LeaP Success

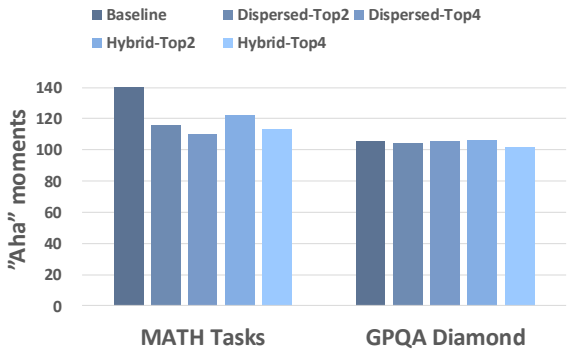
Problem:
"Each vertex of a regular octagon is independently colored either red or blue with equal probability. The probability that the octagon can then be rotated so that all of the blue vertices end up at positions where there were originally red vertices is $\frac{m}{n}$, where m and n are relatively prime positive integers. What is $m+n$?"

Model Response:
...
<summarize> In conclusion, the current insights and results I've gathered are as follows: - The problem involves two configurations of circles arranged in triangle ABC: eight circles of radius 34 and 2024 circles of radius 1,
...
- After extensive exploration and considering peer insights, the most consistent approach involves setting up equations based on the path lengths required by both configurations and solving for the inradius using proportional reasoning, leading to an inradius of ($\frac{384}{5}$),(**resulting in** ($m+n=384+5=389$...)</summarize> <peer_summaries>Peer 1: "To summarize, based on my previous reasoning, I have currently found
...
Thus, GCD is 1, making $\frac{68,800}{2007}$ already in simplest terms. Adding numerator and denominator gives $68,800+2007=70,807$. However, this contrad..." Peer 2: "In short, my current conclusions are that the problem involves arranging circles in geometric progression along an angle bisector of triangle ABC's vertex B.
...
The relationship $2024=8\times 253$ hints that $k^{2016}=34$ simplifies to $(k^8)^{252}=34$, implying $k^8=\sqrt[252]{34}$. This still leaves the inradius..." Peer 3: "In short, I've currently concluded that by modeling the problem using a geometric progression of radii along the angle bisector at vertex B, and setting up equations for both configurations (8 circles of radius 34 and 2024 circles of radius 1), I derived that the sine of half the angle $\theta/2$ is $\frac{11}{1190}$. Substituting back into the inradius formula yields $R=\frac{192}{5}$, which simplifies to $\frac{192}{5}$. Since 192 and 5 are coprime, the final answer is $m+n=197$. This result aligns with the calculations when assuming the incenter is positioned such that the distance from vertex B to the incenter along the angle bisector equals the distance to the last circle plus its radius." Peer 4: "In conclusion, the current key takeaways and results are as follows: By modeling the problem using geometric progressions for the radii of the circles arranged along the angle at vertex B,
...
the expression for R simplifies to $R=\frac{2\times 34^{253/252}}{34^{1/2016}+1}$. Recognizing that $34^{253/252}=34^{1+1/252}$, this becomes R =..." </peer_summaries>
...
Peer 3 suggests an inradius of 192/5, so 192+5=197. Peer 1 and 2 are getting large fractions. Peer 4's answer is unclear. Given that in the Asymptote example, the inradius was about 0.45 for small circles, and the problem has 2024 circles of radius 1,
...
which is not a fraction. Alternatively, the answer is 34/253 +1/2024? No. Alternatively, the problem is a trick question where the inradius is simply the sum of the radii multiplied by the number of circles divided by some number. Alternatively, the inradius is the solution to the equation: 8*(2*34) =2024*(2*1) Which is 544 =4048 → impossible. Thus, the only plausible answer is Peer3's 192/5, which is 38.4.
...
Given that 384/5 is 76.8, and the path length for first case is 544, which is 7*68 + 68=544, and 544/7=77.714. Close to 76.8. Perhaps Peer1's calculation is correct. Alternatively, the inradius is 544/7=77.714= 544/7=77.714285..., which is 544 divided by7 is 77.714. 544 divided by7:7×77=539, so 544/7=77+5/7=77.714285... Thus, as a fraction, 544/7, which reduces to 544/7 (since 544 and7 are coprime). Thus, m+n=544+7=551.
...
then r=34 +476 sinθ To also satisfy the second configuration: r=1 +4046 sinθ Thus, 34+476 sinθ=1+4046 sinθ 33= 3570 sinθ sinθ=33/3570=11/1190=0.0092 Thus, sinθ=0.0092 Then, r=1 +4046*(11/1190)=1 + (4046×11)/1190 Calculate numerator:4046×11=44506 Divide by1190: 44506/1190 =37.4 Thus, r =1+37.4=38.4=384/10=192/5=38.4 Ah, so this gives r=192/5=38.4 Thus, m=192, n=5, so m+n=197 This matches Peer3's answer. Therefore, the inradius is 192/5. The steps: Assuming the progression ratio k=1 (even though impossible), then the inradius is derived as above, leading to sinθ=11/1190, and substituting back gives r=192/5. **Thus, the answer is 192+5=197. **Final Answer**** $\boxed{197}$) ...

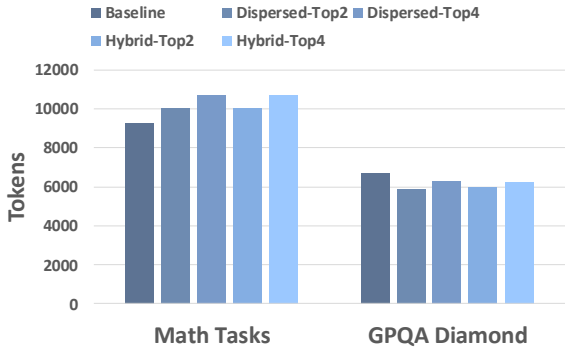
Figure 21: The case study that the model learn from peers successfully, which lead to a correct final answer.



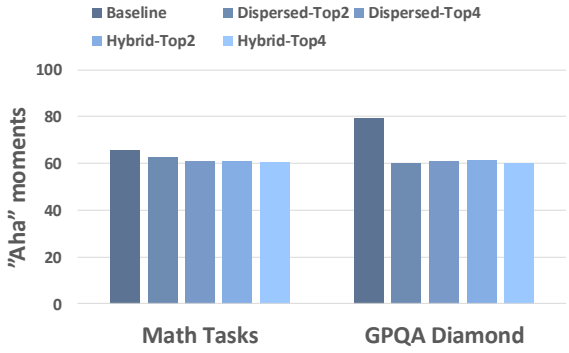
(a) Number of Tokens (QwQ-32B)



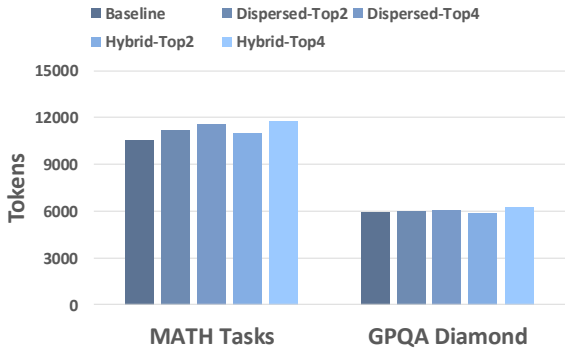
(b) Number of "Aha" moments (QwQ-32B)



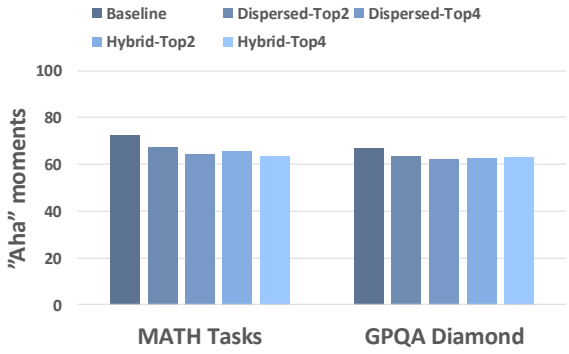
(c) Number of Tokens (DeepSeek-7B)



(d) Number of "Aha" moments (DeepSeek-7B)



(e) Number of Tokens (DeepSeek-14B)



(f) Number of "Aha" moments (DeepSeek-14B)

Figure 22: We illustrate the average number of tokens and "Aha" moments on QwQ-32B (top), DeepSeek-R1-Distill-Qwen-7B (middle), and DeepSeek-R1-Distill-Qwen-14B (bottom). Our method produces a comparable number of tokens to the baseline, while yielding fewer "Aha" moments across all models.